

Sveučilište Jurja Dobrile u Puli
Fakultet ekonomije i turizma
«Dr. Mijo Mirković»

NIKOLINA CINDRIĆ

**RAZVOJ DINAMIČKIH WEB APLIKACIJA
UPORABOM PHP-a I AJAX-a**

Diplomski rad

Pula, 2017.

Sveučilište Jurja Dobrile u Puli

Fakultet ekonomije i turizma

«Dr. Mijo Mirković»

NIKOLINA CINDRIĆ

**RAZVOJ DINAMIČKIH WEB APLIKACIJA
UPORABOM PHP-a I AJAX-a**

Diplomski rad

JMBAG: 0303040428, redoviti student

Studijski smjer: Poslovna informatika

Predmet: Dinamičke web aplikacije

Znanstveno područje:

Znanstveno polje:

Znanstvena grana:

Mentor: prof. dr. sc. Mario Radovan

Pula, rujan 2017.



IZJAVA O AKADEMSKOJ ČESTITOSTI

Ja, dolje potpisana Nikolina Cindrić, kandidat za magistra poslovne ekonomije ovime izjavljujem da je ovaj Diplomski rad rezultat isključivo mogega vlastitog rada, da se temelji na mojim istraživanjima te da se oslanja na objavljenu literaturu kao što to pokazuju korištene bilješke i bibliografija. Izjavljujem da niti jedan dio Diplomskog rada nije napisan na nedozvoljen način, odnosno da je prepisan iz kojega necitiranog rada, te da ikoji dio rada krši bilo čija autorska prava. Izjavljujem, također, da nijedan dio rada nije iskorišten za koji drugi rad pri bilo kojoj drugoj visokoškolskoj, znanstvenoj ili radnoj ustanovi.

Student

U Puli, _____, _____ godine



IZJAVA o korištenju autorskog djela

Ja, Nikolina Cindrić dajem odobrenje Sveučilištu Jurja Dobrile

u Puli, kao nositelju prava iskorištavanja, da moj diplomski rad pod nazivom Razvoj dinamičkih web aplikacija uporabom PHP-a i AJAX-a koristi na način da gore navedeno autorsko djelo, kao cjeloviti tekst trajno objavi u javnoj internetskoj bazi Sveučilišne knjižnice Sveučilišta Jurja Dobrile u Puli te kopira u javnu internetsku bazu završnih radova Nacionalne i sveučilišne knjižnice (stavljanje na raspolaganje javnosti), sve u skladu sa Zakonom o autorskom pravu i drugim srodnim pravima i dobrom akademskom praksom, a radi promicanja otvorenoga, slobodnoga pristupa znanstvenim informacijama.

Za korištenje autorskog djela na gore navedeni način ne potražujem naknadu.

U Puli, _____ (datum)

Potpis

Sadržaj

1. Uvod.....	1
2. Web aplikacije	2
2.1. Vrste aplikacija	4
3. Modeli i metode načina izrade web aplikacija	8
3.1. Model vodopada.....	8
3.2. Model evolucijskog razvoja.....	9
3.3. Model inkrementalnog razvoja.....	10
3.4. Model formalnog razvoja	10
3.5. Model grafičkog razvoja.....	11
3.6. Model usmjeren na ponovnu uporabu	12
4. Skriptni jezici	14
4.1. PHP	15
4.1.1. Najpoznatije aplikacije napravljene koristeći PHP.....	17
4.1.2. Novi trendovi i okviri PHP-a	18
4.1.3. MySQL.....	21
4.2. AJAX	25
5. Primjer aplikacije.....	31
5.1. Baza podataka aplikacije Medias	32
5.2. Dinamička web aplikacija Medias.....	36
5.2.1. Početna stranica	36
5.2.2. Registracija	37
5.2.3. Prijava.....	38
5.2.4. Medias	39
5.2.5. Unos pacijenata	41
5.2.6. Obrada.....	42
5.2.7. Popis pacijenata.....	43
5.2.8. Popis aktivnih pacijenata	44
5.2.9. Izvještaj po pacijentima	45
5.2.10. Izvještaj po zaposlenicima	46
5.2.11. Izvještaj po DTP-u	47
5.2.12. Unos zaposlenika	48
5.2.13. Popis zaposlenika.....	50
5.2.14. Praćenje rada	51

5.2.15. MediasK.....	52
6. Zaključak.....	54
Literatura	55
Prilozi.....	57
Sažetak.....	59
Summary	60

1. Uvod

U početcima World Wide Weba korisnicima su dostupne bile samo informacije koje su se nalazile na statičkim dokumentima poznatim kao web stranice. Proces pružanja informacija bio je jednosmjernan od poslužitelja prema pregledniku, odnosno poslužitelj bi postavio niz informacija na web stranicu koje je preglednik isključivo mogao čitati. Danas je, pak, World Wide Web potpuno drugačijeg sadržaja; više od 80% prostora na webu čine web aplikacije koje omogućavanjem dvosmjernog toka odnosno interakcije poslužitelja i preglednika olakšavaju gotovo svaku korisnu funkciju koja se može provesti online. Upravo zbog velike konkurencije, ali i današnjega užurbanog života, svim web developerima u interesu je stvoriti što sigurniju, lakšu za uporabu, ali ujedno i što učinkovitiju web aplikaciju. Kako bi to ostvarili, koriste se raznim skriptnim jezicima, a kroz ovaj rad prikazati će se razvoj web aplikacije uporabom skriptnih jezika PHP- a i AJAX- a.

2. Web aplikacije

Web aplikacije su klijent-server aplikacije kod kojih klijent odnosno korisnik putem web preglednika npr. „Google Chrome“, „Firefox“, „Opera“ i dr. od servera npr. „Windows server“, „Linux server“ i sl. traži neki sadržaj ili izvršenje određene operacije. Prilikom dolaska na stranicu poslužitelj stranicu poslužuje s malim brojem podataka, no kako tokom uporabe aplikacije korisnik komunicira s poslužiteljem slanjem zahtjeva, server reagira na te zahtjeve i šalje poruke koje uzrokuju ažuriranje trenutne stranice s dodatnim podacima koje je korisnik zatražio. U starom server-driven pristupu poslužitelj je omogućio da cijela stranica generira nove podatke, dok moderni klijent-server pristup omogućuje pružanje resursa na dijelu stranice.¹ Neke od najznačajnijih prednosti i nedostataka suvremenih web aplikacija prikazani su u tablici 1.

<i>PREDNOSTI</i>	<i>NEDOSTATCI</i>
<ul style="list-style-type: none">• Ne zahtijeva složenu proceduru implementacije niti procedure nadogradnje	<ul style="list-style-type: none">• Sigurnost povjerljivih privatnih i osjetljivih informacija
<ul style="list-style-type: none">• Mogućnost rada više korisnika bez smanjenja performansi	<ul style="list-style-type: none">• Brzina rada ovisi o brzini mrežne povezanosti sa poslužiteljem
<ul style="list-style-type: none">• Širok raspon platformi i razvojnih alata	<ul style="list-style-type: none">• Obavezno postojanje internetske mreže
<ul style="list-style-type: none">• Rade bez obzira na OS	<ul style="list-style-type: none">• Vlasnik aplikacije ima kontrolu nad podacima
<ul style="list-style-type: none">• Može se pristupiti s bilo kojeg uređaja (nije potrebna instalacija)	<ul style="list-style-type: none">• Nedovoljna usuglašenost web standarada

Tablica 1: Prednosti i nedostaci web aplikacija

izvor: Horizont, Web aplikacije: prednosti i nedostaci, dostupno na: <http://www.horizont.com.hr/web-aplikacije---prednosti-i-nedostaci-3-blog>

¹ C.Saternos, *Client-Server Web Applications with Javascript and Java:Rich, Scalable and Restful*, O'Reilly Media, inc, 2014., str.3., dostupno na: https://books.google.hr/books?id=3b40AwAAQBAJ&printsec=frontcover&dq=web+applications+are+client-server&hl=hr&sa=X&ved=0ahUKEwj_r_biJtorTAhWiKJoKHeLJAIYQ6AEIPTAF#v=onepage&q=web%20applications%20are%20client-server&f=false, (pristpljeno: 5.4.2017.)

Danas korisnici putem web aplikacija mogu vršiti registracije i prijave u različite vrste sustava u privatne ili poslovne svrhe, provoditi financijske transakcije koristeći web aplikacije banaka koje omogućavaju internet bankarstvo, pretraživati web putem web preglednika kao što su „Google“ i „Yahoo“ ili pak pretraživati brojne baze članaka, knjiga, glazbe i sl., osim navedenog korisnici danas putem web aplikacija mogu kupovati ili prodavati, društveno se povezivati putem web aplikacija kao što su „Facebook“, „Twitter“ i sl. te kontaktirati putem raznih mail servisa. Razvojem web aplikacija omogućeno je da se sadržaj, koji se generira, prilagođava interesima i potrebama svakog pojedinog korisnika. To je omogućio trend u World Wide Web tehnologiji pod nazivom WEB 2.0. koji je u potpunosti promijenio način na koji ljudi danas posluju, komuniciraju ili dijele informacije između sebe. Osnovna karakteristika ovog trenda su interaktivne odnosno dinamičke aplikacije koje korisniku omogućavaju sudjelovanje u doprinosu, organizaciji i stvaranju sadržaja koji im se prikazuje. Prikaz socioloških promjena može se prikazati putem danas najposjećenijih web aplikacija kao što su „Facebook“, „YouTube“, „Wikipedia“ i „Gmail“. „Facebook“ je u početku stvoren kako bi omogućio studentima da ostanu u kontaktu sa svojim prijateljima, dok danas ova društvena mreža ima nekoliko milijuna korisnika koji postavljaju slike iz svojeg života i razne statuse kojima javno objavljuju svoja mišljenja. Svaki korisnik može sam odabrati koje će prijatelje prihvatiti i pratiti, što samo po sebi definira sadržaj koji će se korisniku prikazati na početnoj stranici „Facebooka“. Također, svaki korisnik može sam definirati svoje podatke i odrediti koje podatke želi podijeliti s drugima, čime definira sadržaj koji će se o njemu prikazati njegovim „Facebook“ prijateljima. Wikipedia je, pak, online enciklopedija koja se može dopunjavati i ispravljati od strane svakog korisnika interneta što omogućuje da korisnici sami mogu definirati sadržaj koji će se pojaviti za traženi pojam.² Web aplikacije „Medias“ koja je nastala za potrebe ovog diplomskog rada također je moguće sagledati u kontekstu web 2.0. Svaki vlasnik ustanove za pružanje kućne njege sam definira svoje zaposlenike, pacijente te njihove obrade, što znači da

² Shelly, G.B., Fraydenberg, M., *Web 2.0.: Concepts and Applications*, Cengage Learning, 2010., dostupno na: <https://books.google.hr/books?id=WYSP0ll6ftcC&printsec=frontcover&dq=web+2.0.&hl=hr&sa=X&ved=0ahUKEwtiriDzbzTAhUHQBoKHeFeApUQ6AEIQDAE#v=onepage&q=web%202.0.&f=false>, (pristupljeno : 26.4.2017.)

se svakom vlasniku ustanove koji je registriran u sustav aplikacije „Medias“ prikazuje različiti sadržaj, tj. različiti podatci na stranicama aplikacije.

Osim što se teži da aplikacija bude u kontekstu trenda Web 2.0. poželjno je da aplikacija ima osjetljivi dizajn. Osjetljivi dizajn jest metoda koja omogućava da sav sadržaj bude na stranici, ali da se optimizira ovisno o veličini uređaja odnosno ekrana. Tako će sadržaj koji se nalazi na stranici biti isti, ali drugačije pozicioniran i na desktopu osobnog računala ali i na ekranu pametnog telefona tako da bude prilagođen svakome ovisno o njegovoj veličini ekrana. Kako bi se omogućio osjetljivi dizajn web aplikacije, može se koristiti nekoliko načina, no prilikom izrade web aplikacije „Medias“ korišten je Bootstrap HTML i CSS okvir. Bootstrap je proizvod otvorenog koda koji su razvili Mark Otto i Jacob Thornton za potrebe Twittera 2011. godine. Bootstrap koristi CSS i JavaScript dodatke te ikone koje su prilagođene formama i tipkama.³

2.1.Vrste aplikacija

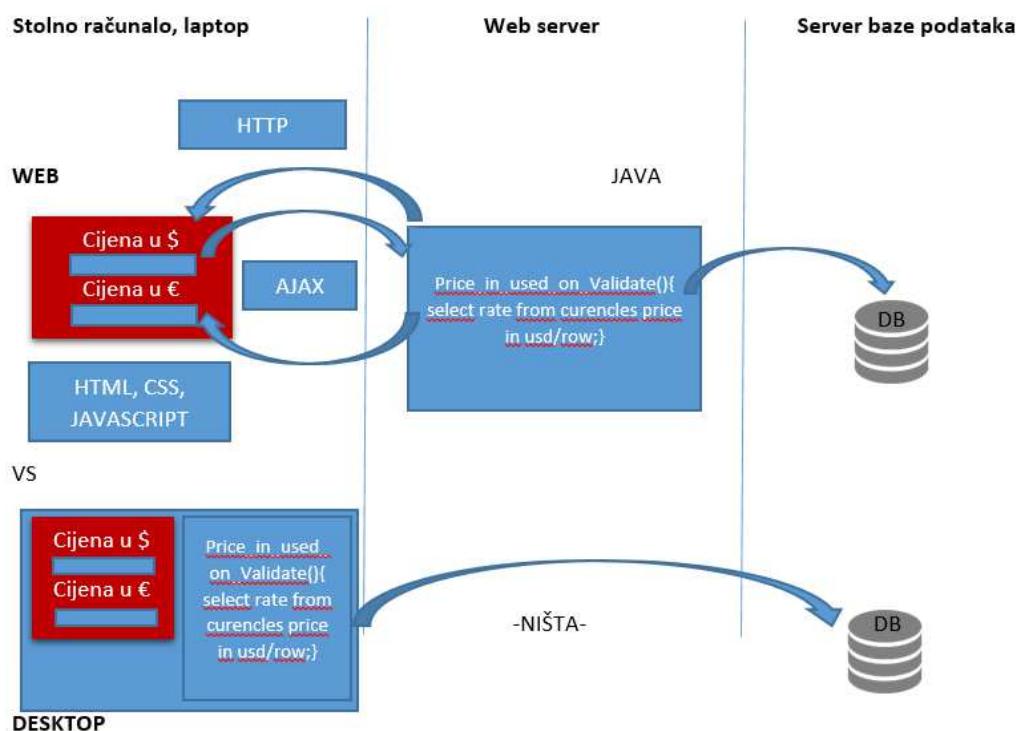
Kada govorimo o vrsti aplikacija, tada možemo govoriti o klasičnim desktop aplikacijama, sve popularnijim dinamičkim web aplikacijama i aplikacijama prilagođenim mobilnim uređajima.

Klasične desktop aplikacije ograničene su na hardverske uređaje na kojima su instalirane te njihova primjena ovisi o operacijskom sustavu koji pokreće taj hardverski uređaj. Svaki korisnik mora instalirati novu verziju aplikacije te može imati dodatne zahtjeve vezane uz memoriju ili samu nadogradnju hardvera. Današnje desktop aplikacije preuzele su neke pozitivne strane web aplikacija, ali i korištenje nekih skriptnih jezika kao npr. HTML-a, JavaScripta i sl.

Dinamičkim web aplikacijama korisnici mogu pristupiti s bilo kojeg hardverskog uređaja koji ima internetsku vezu. Korisnik pristupa aplikaciji s bilo kojeg preglednika

³ Spurlock, J., Bootstrap: Responsive Web Development, O'Reilly Media inc., 2013. , dostupno na: <https://books.google.hr/books?id=LZm7Cxgi3aQC&printsec=frontcover&dq=web+applications+responsive+design+bootstrap&hl=hr&sa=X&ved=0ahUKEwjV4syNncLTAhVQrRQKHcisD2YQ6AEIZTAJ#v=onepage&q&f=false> , (pristupljeno: 26.4.2017.)

te radi s dostupnim informacijama. Sve više web aplikacija veliki napredak vidi u offline verziji aplikacije koja će omogućiti korisnicima korištenje aplikacije i kada internetska veza nije dostupna. Razlika između desktop i web aplikacija može se vidjeti na slici 1.

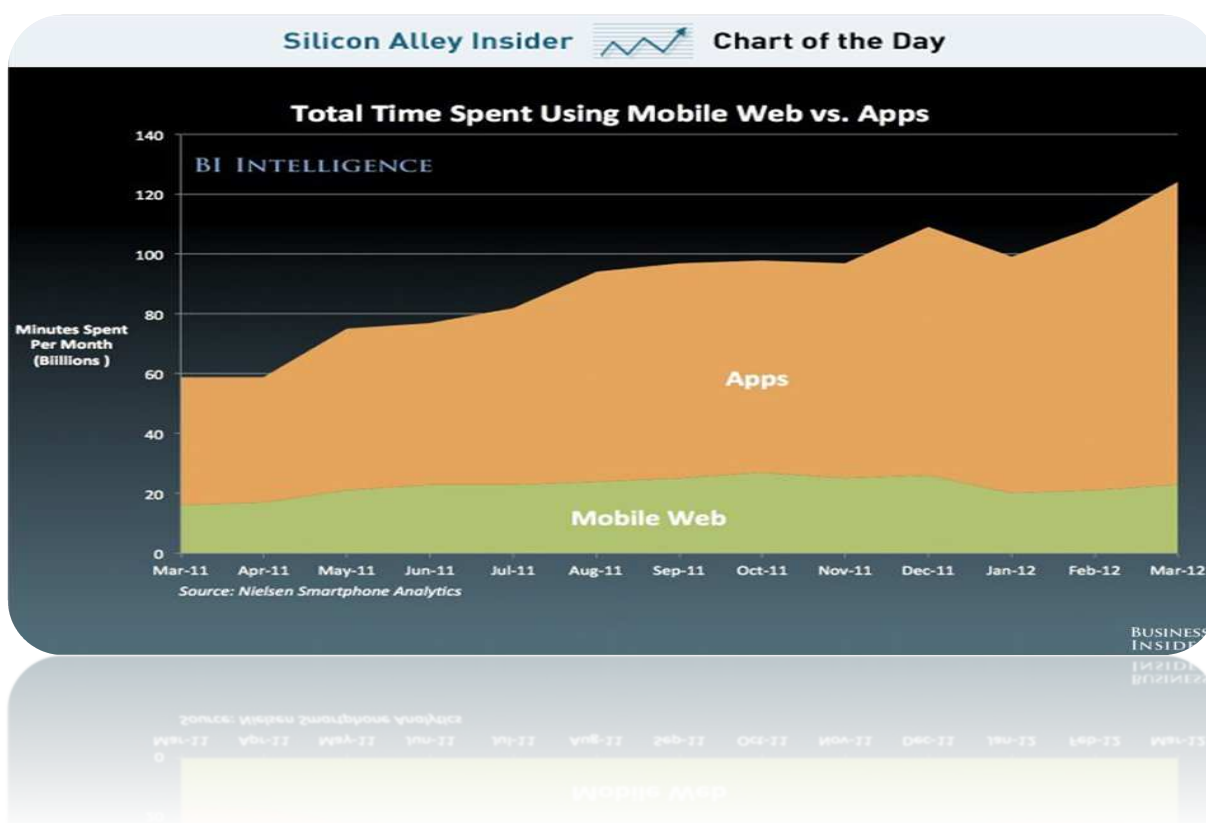


Slika 1. Razlika između desktop aplikacija i web aplikacija

izvor: Coderanch, *Difference between desktop application or web application*, dostupno na: <https://coderanch.com/t/636684/java/Difference-Web-application-desktop-application>

Dinamičke web aplikacije prilagođene mobilnim uređajima moraju zadovoljiti neke zahtjeve dizajna osjetljivog na dodir. Dizajn osjetljiv na dodir podrazumijeva stvaranje aplikacije koja će jednako izgledati na hardverskim uređajima različite veličine te koja će imati iste funkcionalnosti neovisno radi li se o osobnim računalima i laptopima ili o pametnim telefonima, koji imaju zaslon osjetljiv na dodir. Danas razlikujemo native ili tzv. mobilne aplikacije i hibridne mobilne aplikacije.⁴ Posljednjih nekoliko godina sv je veća uporaba mobilnih uređaja, a samim time i mobilnih aplikacija. Odnos korištenja mobilnih aplikacija i web aplikacija na mobilnim uređajima u 2012. godini prikazan je na slici 2.

⁴ Segue Technologies, "Desktop vs. Web applications: A Deeper Look and Comparison", dostupno na: <http://www.seguetech.com/desktop-vs-web-applications/>, (pristupljeno: 5.4.2017.)



Slika 2. Prikaz korištenosti web aplikacija u odnosu na mobilne aplikacije

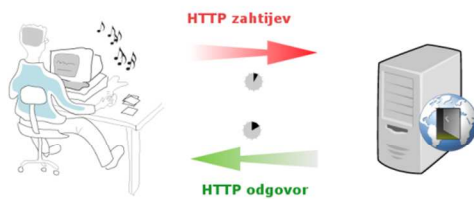
izvor: *Beyond PLM, Mobile PLM; Apps vs. Mobile browser*, dostupno na:
<http://beyondplm.com/2012/10/17/mobile-plm-apps-vs-mobile-browsers/>

Također moramo razumjeti razliku između statičnih web stranica i dinamičkih web aplikacija koja je prikazana na slici 3. Za razliku od dinamičkih web aplikacija, statične web stranice ne mijenjaju svoj sadržaj, osim ako se sadržaj stranice promijeni direktno u kodu. Statičke stranice sadrže samo tekst koji je upisan u kodu, slike ili animacije, no ne sadrže forme ili slične elemente koji korisniku omogućavaju interakciju ili povezivanje s bazom podataka. Dakle, korisnik pomoću statičke web stranice može vidjeti samo tekst, npr. podatke o firmi i sl. dok putem dinamičke web aplikacije korisnik može sam „komunicirati“ s web aplikacijom.⁵

⁵J.W.Lanz, The Essential Attorney Handbook for Internet Marketing, Search Engine Optimization, and Website Deve, Dostupno na:
https://books.google.hr/books?id=DvBYf16u_DYC&pg=PA98&dq=dynamic+vs+static+website&hl=hr&sa=X&ved=0ahUKEWjKI7nSrIPUAhVIBMAKHTn5Ca0Q6AEIIDA#v=onepage&q=dynamic%20vs%20static%20website&f=false

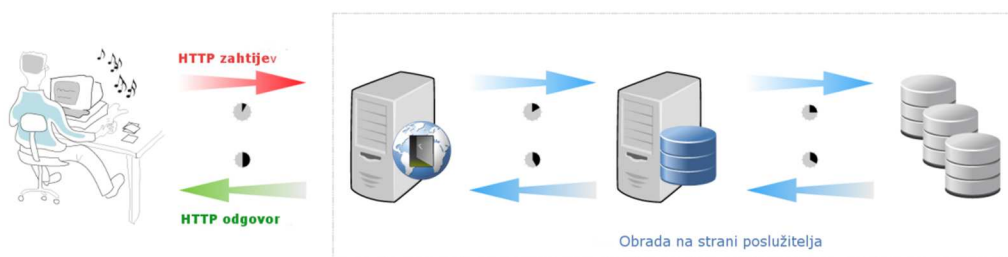
Statična web stranica

Šema A



Dinamička web stranica

Šema B



Slika 3. Dinamičke web aplikacije vs. statičke web stranice

izvor: Gitlab, SSgs part1: Static vs. Dynamic Websites, dostupno na:
<https://about.gitlab.com/2016/06/03/ssg-overview-gitlab-pages-part-1-dynamic-x-static/>

3. Modeli i metode načina izrade web aplikacija

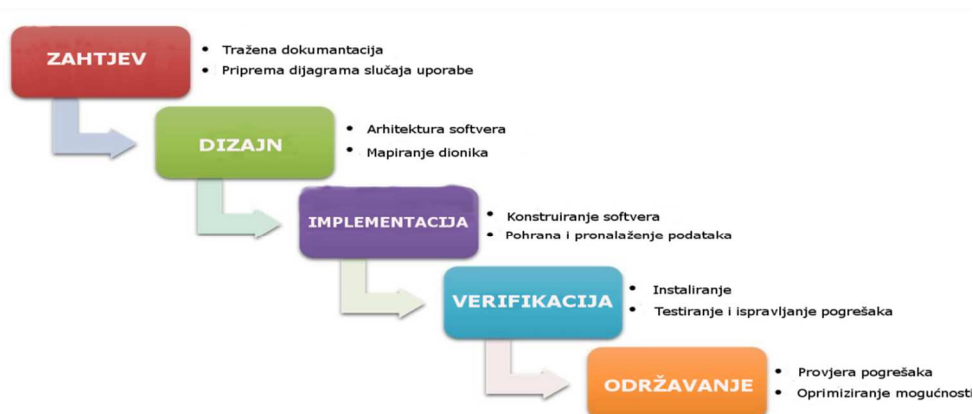
Modeli izrade softverskog procesa, u ovom slučaju dinamičkih web aplikacija, predstavljaju njegov idealizirani prikaz kojim se definira poželjan način djelovanja tog procesa te međusobnog povezivanja njegovih osnovnih aktivnosti. Svaki od modela izrade softverskog procesa sastoji se od istih aktivnosti, a razlikuju se na temelju odvijanja pojedinih aktivnosti, međusobnog povezivanja osnovnih aktivnosti te naglaska na pojedine aktivnosti. Osnovne aktivnosti prisutne kod svakog modela su:

- analiziranja korisničkih zahtjeva i utvrđivanja funkcionalnosti sustava
- oblikovanja građe sustava i njegovih dijelova te dizajniranja sučelja
- implementacija, tj. programiranje samog softverskog procesa
- verifikacija i validacija
- održavanje i nadogradnja nakon uvođenja u uporabu.

Razlikujemo šest vrsta modela izrade softvera: Model vodopada, Model evolucijskog razvoja, Model inkrementalnog razvoja, Model formalnog razvoja, Model grafičkog razvoja i Model usmjeren na ponovnu uporabu.⁶

3.1. Model vodopada

Aktivnosti kod ove vrste modela izrade softverskog procesa teku sekvencijalno kao faze jedna iza druge, a svaka faza započinje rezultatom prethodne faze što je vidljivo na slici 4.



Slika 4: Model vodopada

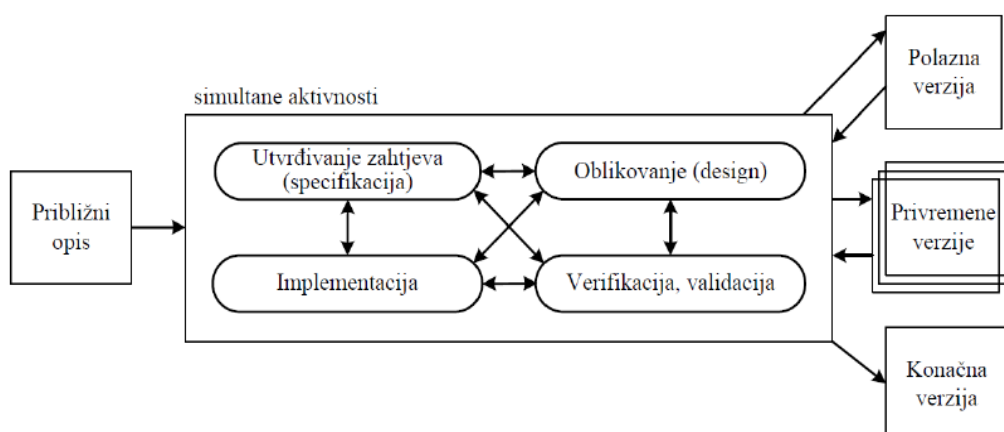
⁶ R.Manger, *Softversko inženjerstvo*, Prirodoslovno matematički fakultet u Zagrebu, 2005., str.2.

izvor: *Presentation proces, Smart waterfall Model*, dostupno na: <http://www.presentation-process.com/smartart-waterfall-model-video.html>

Osnovne pogodnosti ovog modela su mogućnost detaljnog planiranja cjelokupnog sustava i podjela na veliki broj suradnika te jednostavno definiranje faze u kojoj se proces nalazi. Mane ovog modela su nemogućnost razdvajanja pojedinih faza u praksi, razvijanje procesa u nepoželjnom obliku zbog nepoželjnih vraćanja na prethodne faze ili „zamrzavanja faza“, zbog sporosti sustava proizvod može biti neažuriran. Usprkos navedenim prednostima i manama ovaj se model često koristi kod razvoja velikih sustava s velikim brojem programera.⁷

3.2. Model evolucijskog razvoja

Aktivnosti kod modela evolucijskog razvoja provode se simultano bez mogućnosti razdvajanja pojedinih aktivnosti (slika 5.). Na temelju približnog opisa problema izrađuje se početna verzija sustava, koja se na temelju primjedaba od strane korisnika dorađuje u iteracijama sve dok se ne dobije konačan proizvod. Pogodnosti ovog modela su u tome što omogućava brzi odgovor na zahtjeve korisnika te mogućnost razvoja sustava uz nejasne zahtjeve. Mane ovog modela su nemogućnost definiranja postotka napravljenog posla, što menadžerima otežava njihov posao, loša struktura konačnog sustava uzrokovana stalnim promjenama. Za uspješan gotov sustav neophodni su posebni alati i natprosječni softverski inženjeri. Ovaj model koristi se za razvoj web mjesta, male sustave ili sustave s nejasnim zahtjevima.⁸



Slika 5: Model evolucijskog razvoja

⁷ ibidem, str. 3.

⁸ ibidem, str. 4.

3.3. Model inkrementalnog razvoja

Model inkrementalnog razvoja spoj je prethodnih dvaju modela tako da se proces razvoja provodi kroz iteracije, no ne dorađuje se prethodna iteracija na temelju primjedaba korisnika, već se dodaje novi dio prethodnoj iteraciji tzv. inkrement. Razvoj svakog inkrementa provodi se na temelju modela vodopada (slika 6.).



Slika 6: Model inkrementalnog razvoja

izvor: R.Manger, Softversko inženjerstvo, Prirodoslovno matematički fakultet u Zagrebu, 2005.

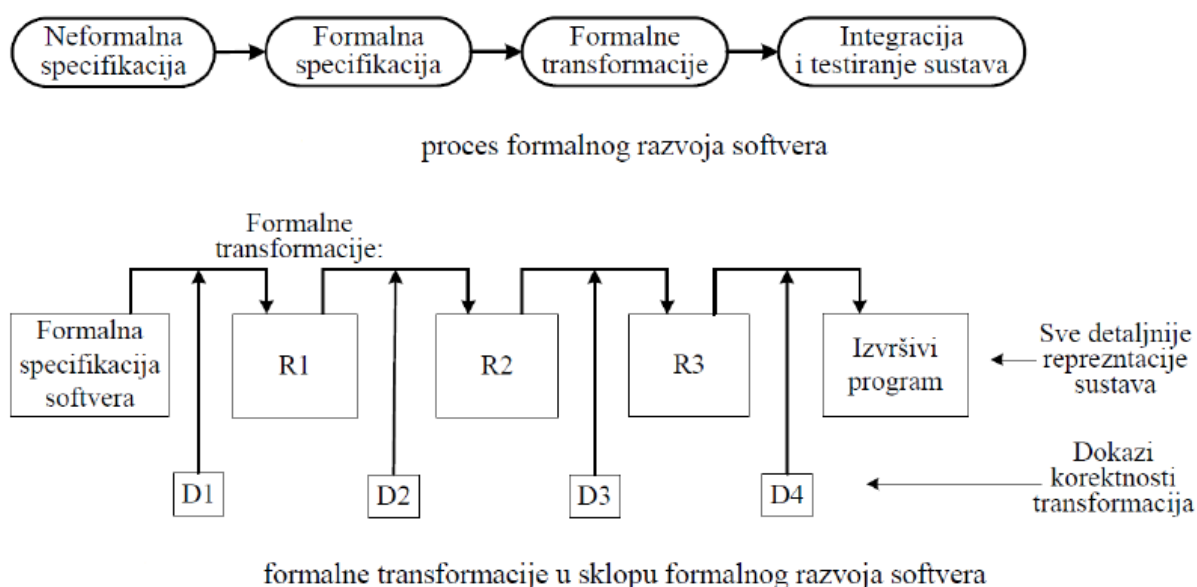
Pogodnosti ovog modela su u tome što omogućava definiranje koji je inkrement završen, što olakšava posao menadžerima te se prvi inkrement koji zadovoljava osnovne potrebe korisnika relativno brzo razvija. Mane ovog modela su otežana podjela korisničkih zahtjeva u smislene inkremente i otežano definiranje zajedničkih modula koji moraju biti implementirani u početnom inkrementu. Svi generički softverski paketi, uključujući i Microsoft Office, razvijeni su na temelju ovog modela, te svakim novim inkrementom donose nove mogućnosti korisnicima.⁹

3.4. Model formalnog razvoja

Kod modela formalnog razvoja zahtjevi korisnika precizno se definiraju na formalan način, korištenjem nedvosmislene matematičke notacije, potom se formalna

⁹ loc.cit.

specifikacija nizom koraka, koji čuvaju korektnost, transformira do programa kao što je prikazano na slici 7. Pogodnosti ovog modela su mogućnost automatizacije svih faza razvoja nakon izrade formalne specifikacije, nema potrebe za testiranjem jer se matematički može odrediti točnost sustava, te rješenje ne ovisi o platformi. Mane ovog modela su te što je za izradu formalne specifikacije potrebno puno truda i znanja, korisnici ne mogu pratiti razvoj te postojeći specifikacijski jezici nisu pogodni za interaktivne sustave. U praksi se koristi jako malo i isključivo za sustave koji zahtijevaju veliku sigurnost i pouzdanost, no ako se razviju bolji specifikacijski jezici i bolji alati za automatske transformacije, moguća je veća uporaba ovog modela u budućnosti.¹⁰



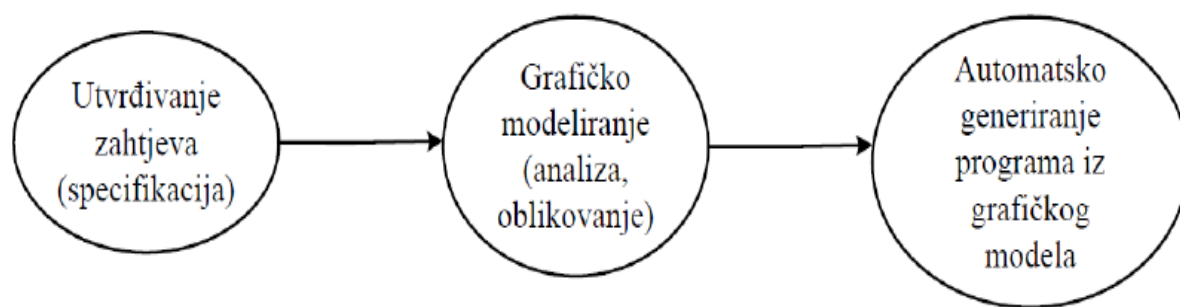
Slika 7: Model formalnog razvoja

izvor: R.Manger, Softversko inženjerstvo, Prirodoslovno matematički fakultet u Zagrebu, 2005.

3.5.Model grafičkog razvoja

Aktivnosti kod modela grafičkog razvoja teku isto kao i kod modela formalnog razvoja, no umjesto formalne specifikacije kod ovog modela izrađuje se skup dijagrama izrađenih UML grafičkim jezikom za modeliranje, kojima se prikazuje građa i ponašanje sustava (slika 8.). Modeli se na temelju pojedinih alata mogu automatski pretvarati u programe.

¹⁰ ibidem, str.5.



Slika 8: Model grafičkog razvoja

izvor: R.Manger, Softversko inženjerstvo, Prirodoslovno matematički fakultet u Zagrebu, 2005.

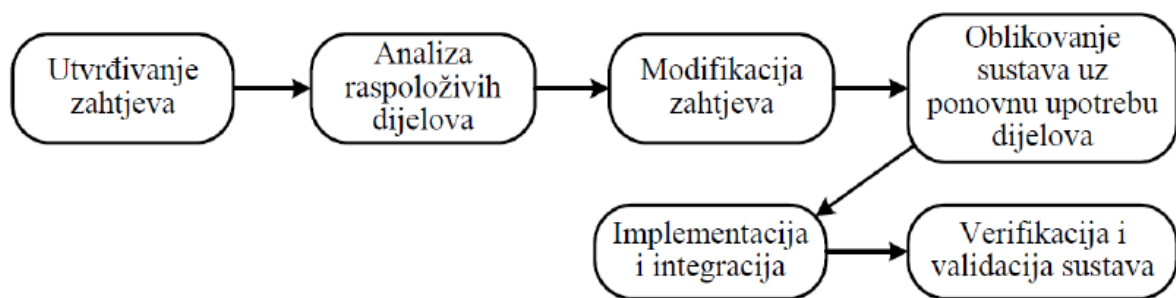
Kao i kod formalnog modela i kod modela grafičkog razvoja pogodnosti su: mogućnost automatizacije svih faza razvoja nakon izrade formalne specifikacije, nema potrebe za testiranjem jer se matematički može odrediti točnost sustava te rješenje ne ovisi o platformi, ali i jednostavnije korištenje i mogućnost generiranja dijela programskog koda na temelju UML dijagrama. Osnovna je mana ovog modela što je upotrebljiv u isključivo vrlo jednostavnim primjerima, što se pokušava riješiti uvođenjem posebnih jezika kao npr. Object Constraint Language, koji služi za specifikaciju ograničenja koja putem UML dijagrama nisu vidljiva.¹¹

3.6.Model usmjeren na ponovnu uporabu

Model usmjeren na ponovnu uporabu temelji se na ideji da se novi sustavi razvija spajanjem postojećih gotovih i upotrebljivih softverskih cjelina ili dijelova prije razvijenih sustava. Pogodnosti ovog modela su manja količina softvera koji se treba razviti i smanjenje vremena, troškova i rizika. Mane ovog modela su mogućnost nepotpunog odgovaranja potrebama korisnika zbog kompromisa prilikom izrade specifikacije te mogućnost djelomičnog gubitka kontrole nad evolucijom sustava. S obzirom na veliki broj gotovih rješenja i sve manjeg vremena korisnika, u budućnosti se očekuje kako će ovaj model biti najkorišteniji prilikom izrade softvera.¹² Faze ovog modela prikazane su na slici 9.

¹¹ Ibidem, str. 5.

¹² Ibidem, str. 5.

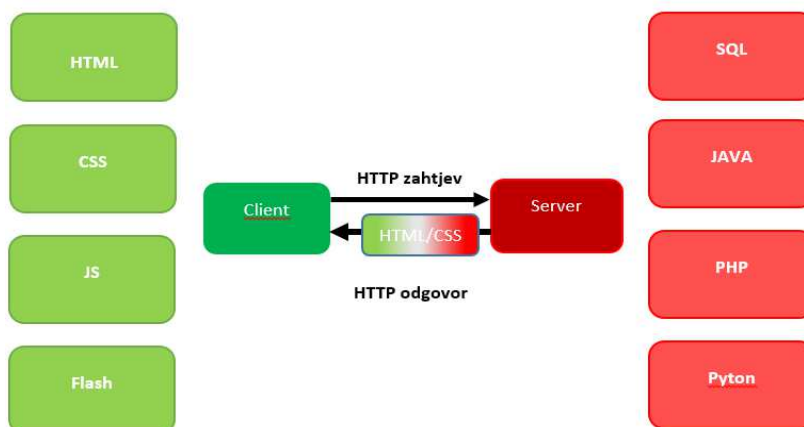


Slika 9. Model usmjeren na ponovnu uporabu

izvor: R.Manger, Softversko inženjerstvo, Prirodoslovno matematički fakultet u Zagrebu, 2005.

4. Skriptni jezici

Skriptni jezici, koji se mogu ugraditi u HTML, koriste se kako bi statičkoj web stranici dodali neku funkcionalnost. Svaka dinamička web aplikacija ima server funkcije i klijent funkcije koje su prikazane na slici 10. Suvremene web aplikacije, iako i dalje sadrže određeni broj statičkih resursa, trude se što veći broj sadržaja generirati dinamički i omogućiti korisnicima brzi odgovor koji je prilagođen svakom pojedinom korisniku. Dinamičko generiranje korisnikova sadržaja provodi se na temelju koda koji se izvodi na poslužitelju i korisnik u većini slučajeva nije svjestan da se ova funkcija odvija. Skriptni jezici koji omogućavaju server funkciju su: PHP, Perl, ASP, JSP, .NET, Python, java i sl. Kroz rad će se prikazati razvoj web aplikacije uporabom PHP skriptnog server-side jezika. Klijent funkcija web aplikacije označava primanje sadržaja koji je korisnik unio te izvođenje akcije koju je korisnik odabrao i prikaz rezultata koje pomoću skriptnih server-side jezika dobiva od strane poslužitelja. U ovom slučaju veliku važnost ima samo korisničko sučelje preko kojega korisnik pristupa web aplikaciji i želi izvršiti željene radnje. Najpoznatiji skriptni jezici koji se koriste za klijent funkcije su HTML, CSS, JavaScript i ActionScript. U ovome radu prikazat će se razvoj web aplikacija uporabom JavaScripta odnosno AJAX-a.¹³



Slika 10: Klijent vs. server skriptni jezici

izvor: Niklas Mathiesen, *Protocols and Web Dynamics*, dostupno na: <http://niklasmathiesen.com/protocols-and-web-dynamics/>

¹³ Stuttard, D. i M.Pinto, *The Web Application Hacker's Handbook: Finding and Exploiting Security Flaws*, 2. Izdanje, O'Reilly, John Wiley & Sons, 2011., Chapter 3: Web Applications Technologies, Web Functionality, dostupno na: <https://www.safaribooksonline.com/library/view/the-web-application/9781118026472/chap03-sec016.html>, (pristupljeno: 23.3.2017.)

4.1.PHP

Rasmus Lerdorf je autor PHP skriptnoga jezika koji je predstavio 1994. godine pod nazivom PHP/FI koji je sadržavao niz Perl skripti koje je Rasmus koristio za praćenje posjeta korisnika njegovog životopisa. S vremenom je cilj postao stvoriti kostur pomoću kojega bi se PHP mogao dodatno nadograđivati i obogaćivati s još alata, no i dalje nije postojala ideja za novi skriptni jezik. Lerdorf je počeo raditi na projektu koji je zahtijevao konsolidiranje podataka s raznih mjesta i njihov prikaz na ugodnom administrativnom sučelju, pa je Lerdorf napisao C implementaciju koja je mogla komunicirati s bazom, ali i omogućiti stvaranje jednostavnih dinamičkih web stranica ili aplikacija. PHP nastavio se razvijati te je drugu verziju PHP/FI-a, koja je svjetlo ugledala 1997. godine, imalo instaliranu više od 50 000 domena. PHP koji je sličan današnjem polazi od treće inačice PHP 3, koju su razvili Andi Gutmans i Zeev Suraski za potrebe projekta na fakultetu. Ova inačica omogućila je korištenje pojedinih ekstenzija kao npr. povezivanje na bazu ili API za komunikaciju, ali i korištenje objektno-orijentirane sintakse. Slijedile su verzije PHP4 i PHP5 koje su omogućile bolju jezgru za objektno orijentirano programiranje.¹⁴ Danas je PHP jedan od najmoćnijih server okvira za izradu web aplikacija (slika 11.).¹⁵ Današnje verzije PHP jezika služe za obavljanje triju osnovnih funkcija:

- skriptiranje sa strane poslužitelja – osnovna funkcija za koju je stvoren PHP jezik jest izrada dinamičkog web sadržaja, te i dan danas najbolje služi upravo za ovu funkciju uz razvoj poboljšanje njegovih funkcija
- naredbeno skriptiranje – slično kao i Perl i PHP može izvoditi skripte iz naredbene linije, te se mogu upotrijebiti kod poslovnih administracijskih sustava kao npr. stvaranje kopija i analiziranje dnevnika događaja
- GUI aplikacija sa strane korisnika – uporabom PHP-GTK moguće je napisati potpuno razvijene GUI aplikacije za uporabu na nekoj platformi.¹⁶

¹⁴ Božajić, I., uvod u PHP i objektno orijentirano programiranje, 2011., Algebra učilište, str. 4.

¹⁵ Lerdorf, R., Programing PHP, 2002., str. 3.

¹⁶ Lerdorf, R., Programing PHP, 2002., str. 1.



Slika 11: Korištenost PHP-a kao server-side skriptnog jezika

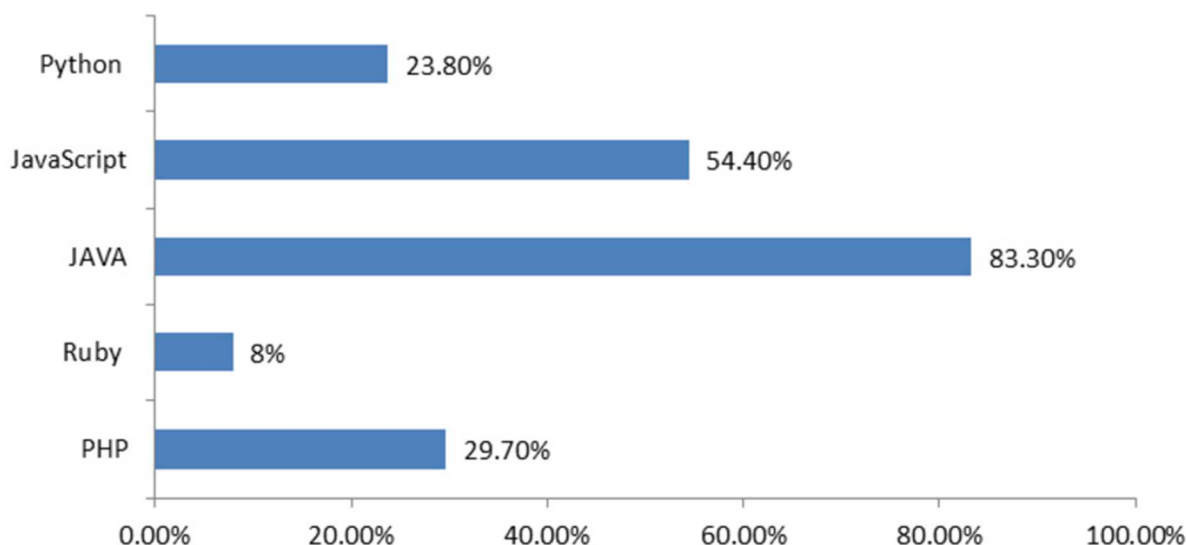
izvor: StoneRiver eLearning, Top 5 Programing Languages Used In Web Developmet, dostupno na: <http://blog.stoneriverelearning.com/top-5-programming-languages-used-in-web-development/>

Na temelju istraživanja provedenog od strane web lokacije W3Tech u svibnju 2012. godine utvrđeno je kako je PHP zastupljen na 78% web lokacija koje su obuhvaćene tim istraživanjem, a koje su činile milijun najposjećenijih web lokacija u 2012. godini, čime je potvrđena široka primjena PHP-a kao skriptnog jezika prilikom izrade web lokacija.¹⁷ Slika 8 prikazuje zastupljenost PHP-a kao server-side skriptnog jezika, gdje je vidljivo kako je PHP uvjerljivo najkorišteniji server-side skriptni jezik. Čak preko 75% web aplikacija koristi PHP kao svoj server-side skriptni jezik.¹⁸ Slijede ga .Net, Javam Ruby te Python kao najmanje korišten server-side skriptni jezik. Ako, pak, pogledamo korištenost PHP općenito u programiranju, stvari nisu iste kao i kod server-side skriptnih jezika. PHP u ovoj situaciji zaostaje za Javom koja je najkorišteniji programski jezik (83,30%) i JavaScriptom koji se koristi u čak 54,40 % programiranja, a što je vidljivo na slici 12.

¹⁷ Lerdorf, R., Programing PHP, 2002., str. 6.

¹⁸ Stone Liver eLearning, "Top 5 Programming Languages Used In Web Development", dostupno na: <http://blog.stoneriverelearning.com/top-5-programming-languages-used-in-web-development/>, (pristupljeno: 23.3.2017.)

Najčešće korišteni programski jezici (% korištenja)



Slika 12: Najkorišteniji programski jezici

izvor: StoneRiver eLearning, Top 5 Programing Languages Used In Web Developmet, dostupno na: <http://blog.stoneriverelearning.com/top-5-programming-languages-used-in-web-development/>

4.1.1. Najpoznatije aplikacije napravljene koristeći PHP

S obzirom na to da je PHP najzastupljeniji server-side skriptni jezik u nastavku su prikazane neke od najpoznatijih web aplikacija koje su napravljene koristeći upravo PHP skriptni jezik.

- „WordPress“ – nastao je zajedničkim radom Matta Mullenwega i Mikea 2003. godine kao nadogradnja cafeloga. „WordPress“ napisan je u PHP-u uz korištenje MySQLa. Od 2017. godine koristi ga 58,7% svih web stranica sa sustavom za upravljanje.¹⁹
- „Facebook“ – najpoznatija socijalna mreža današnjice u svom kodu koristi PHP za prijavu u sustav, prijenos fotografija i videozapisa te slanje serija zahtjeva. „Facebook“ putem PHP-a komunicira sa svojim specifičnim hosting okruženjima i web okvirima.²⁰

¹⁹ WordPress.org, History, dostupno na: <https://codex.wordpress.org/History>, (pristupljeno: 23.3.2017.)

²⁰ Facebook for developers, Facebook SDK v5 for PHP, dostupno na: <https://developers.facebook.com/docs/reference/php/>, (pristupljeno: 23.3.2017.)

- „MediaWiki“ – ovaj OpenSource softver napisan je za potrebe „Wikipedije“ u PHP-u, no danas je u uporabi i za druge neprofitne wiki zaklade.²¹
- „phpMyAdmin“ – „phpMyAdmin“ je alat nastao 1998. godine napisan u PHP-u namijenjen upravljanju administracijom MySQLa putem weba. Omogućuje stvaranje i brisanje baze podataka, kreiranje / poništavanje / mijenjanje tablica, brisanje / uređivati/ dodati polja, te izvršiti bilo koju SQL izjavu.²²
- „Joomla!“ - Nagrađivani web-based sustav za upravljanje sadržajem. Omogućuje pristup sadržaju prednjeg sadržaja i pristup administratorskoj pozadini. Kontrola pristupa skupini omogućava različite razine kontrole sustava za web mjesto i administratora. „Joomla!“ Okvir omogućuje proširenje komponenata (aplikacija) koje se instaliraju, modula (blokovi predloška), jezika, predložaka i mambota (dodataka koji poboljšavaju funkcije sustava). Joomla je jedan od najboljih resursa za učenje PHP-a zahvaljujući izvornom PHP kodu.²³

4.1.2. Novi trendovi i okviri PHP-a

PHP je u 2016. godini dobio svoje novo izdanje u obliku PHP 7.1. gdje je prema mišljenju PHP zajednice vidljiva stabilnost, sazrijevanje i odlučan put naprijed PHP-a. PHP 7.1. riješio je osnovni problem svoje prethodne verzije PHP 7.0., a to je problem s pojedinim tipovima podataka i vraćanjem podataka null tipa. Također, jedno od najvećih prednosti PHP 7.1. su brzina, poboljšana dospijeća PHP knjižnica te okviri. Prema članovima zajednice PHP-a koji su svoje mišljenje dali putem Reddita i Facebooka smatra se kako je PHP uvođenjem PHP 7.1 ušao u potpuno novu eru, te to potvrđuje vjerovanje kako će PHP i dalje ostati jedan od najpopularnijih skriptnih jezika. Kao trendovi PHP-a koji se očekuju u 2017. godini su:

- potpuna dominacija PHP 7.1. u svim aspektima razvoja web stranica i aplikacija, te prestanak podrške za PHP 5.6.

²¹ WikiMedia, dostupno na: <https://www.mediawiki.org/wiki/MediaWiki>, (pristupljeno: 23.3.2017.)

²² Federico Cargnelutti, "20 Most influential Open-Source Web Applications", dostupno na: <https://blog.fedecarg.com/2008/05/22/20-most-influential-open-source-web-applications/>, (pristupljeno: 26.4.2017.)

²³ Joomla!, Documentation, dostupno na: https://docs.joomla.org/PHP_essentials, (pristupljeno: 26.4.2017.)

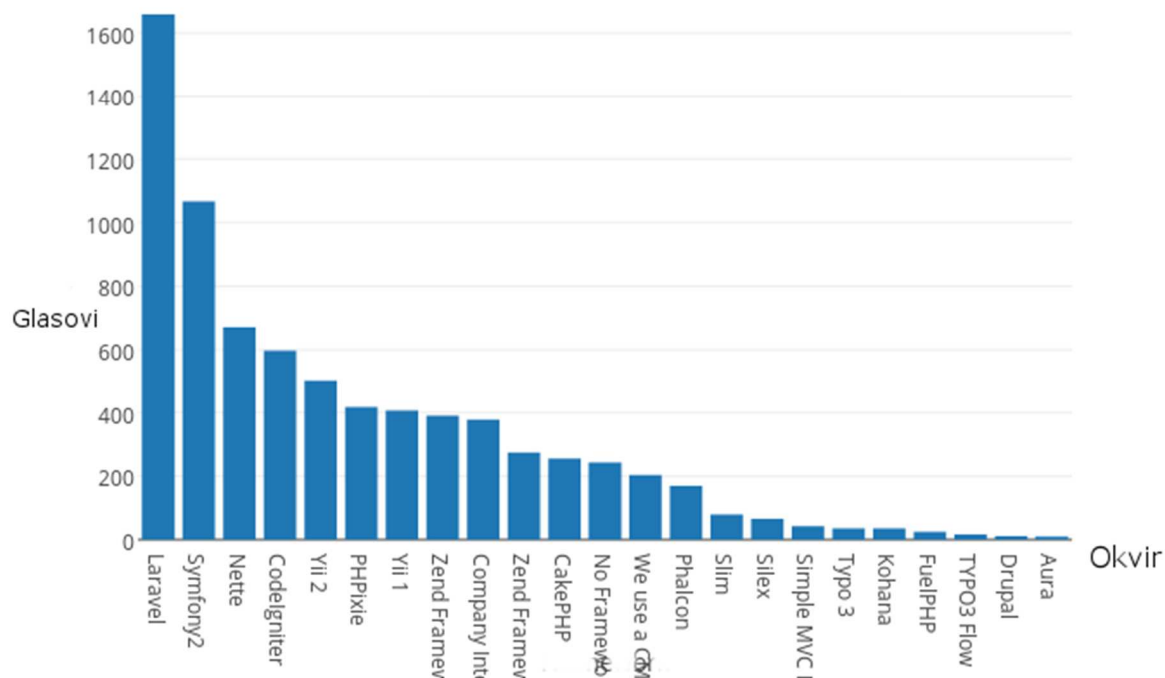
- svi veći PHP okviri prihvatit će PHP 7.1. čime će na raspolaganje dobiti sve njegove važne značajke kao i dio alata
- svi hosting provideri dobit će nadogradnju na PHP 7.1.

Php okviri koriste se prvenstveno kako bi omogućili vrlo jednostavan i učinkovit razvoj web aplikacija, odnosno pomažu web developerima u brzom razvoju web aplikacija, što dovodi do štednje vremena te smanjenja količine ponovljenog kodiranja. Postoje brojni PHP okviri, no neki od najpoznatijih su Symfony, Laravel, Zend Framework, Yii, Silex. Prema mišljenju članova zajednice ne postoji najbolji PHP okvir, u SAD-u najkorišteniji je Laravel zbog svojih iznimnih alata i ekosustava izgrađenog oko njega. U Aziji se najviše koristi Yii PHP okvir koji je prema osnovnim značajkama i performansama nešto iznad svih ostalih PHP okvira. Velika poduzeća za svoju podršku na svjetskoj razini najviše koriste Symfoni.²⁴ Prema istraživanjima u 2016. godini kao PHP okvir, koji se najviše koristio u pojedinačnim projektima kao što je vidljivo na slici 10, daleko najpopularniji je Laravel s preko 2000 glasova, a slijede Symfoni, Nette, Codelgniter i Yii2.²⁵

²⁴ Coludways, „How PHP Will Fare IN 2017?“, dostupno na: <https://www.cloudways.com/blog/php-trends-2017/>, (pristupljeno: 24.3.2017.)

²⁵ SpyreStudio, „Top Trending PHP Frameworks in 2016.“, dostupno na: <http://spyrestudios.com/top-php-frameworks-2016/>, (pristupljeno: 24.3.2017.)

Najpopularniji PHP okvir na radnom mjestu, 2015.



Slika 13: Najpopularniji PHP okviri u 2016. Godini

izvor: Plotly, dostupno na: <https://plot.ly/~swader/8/most-popular-php-framework-at-work-sitepoint-2015/>

U moru Php okvira, kao što je vidljivo i na slici 13., posljednjih godina najveću pozornost web developera privlači php okvir Laravel. Svaki programer koristi onaj Php okvir koji mu prema određenim kriterijima najviše odgovara. U tablici 2. prikazana je razlika između najpopularnijih php okvira Yii-a, Symfoni-a i Laravel-a po pojedinim kriterijima.

Kriteriji	Yii	Symfony	Laravel
Licenca	BSD	MIT	MIT
Zahtjevi	>= PHP 5.4.	>= PHP 5.5.9.	>= PHP 5.5.9.with OpenSSL,PDO, Mbstring, Tokenizer extensions
Internacionalizacija I18N / L10n	Gettext ,Database, PHP array	XML, CSV, PHP	PHP array
Pohrana memorije	APC, database, eAccelerator, File...	HTTP, APC, eAccelerator, Xcache	File System, Database, Memcached, APC, Redis, Xache, Memory(arrays)
Predložak sustava	PHP and Prado's	PHP, Twig	Blade, PHP, Custom

Tablica 2. Razlike između PHP okvira

izvor: Socialcompare, PHP frameworks comparison, dostupno na:

<http://socialcompare.com/en/comparison/php-frameworks-comparison>

4.1.3. MySQL

PHP omogućuje podršku preko dvadeset baza podataka , uključujući i temelj suvremenih dinamičkih web stranica i aplikacija poput MySQL, PostgreSQL i Oracle. U njima se nalaze podatci o nabavi, narudžbama, cijenama, kupcima, bankovnim računima i sl. MySQL najpopularnija je platforma za baze podataka koju koristi PHP.²⁶ MySQL najjednostavnije se može definirati kao softver ili poslužitelj baze podataka kojemu se putem internetske mreže kojemu se može pristupiti korištenjem korisničkom imena i lozinke. Na serveru se može naći više neovisnih baza podataka čijim se podacima može baratati unutar jednog projekta.²⁷

Za korištenje MySQL baza podataka može se instalirati MySQL server ili koristiti online server. Nakon registracije na serveru i izrade baze može se pristupiti izradi tablica pojedine baze podataka koristeći se SQL programskim jezikom za rad s bazama podataka. Kako bi mogli koristiti podatke iz izrađenih tablica u PHP kodu potrebno je stvoriti konekciju na bazu podataka kao što je prikazano na slici 14.

²⁶ Lerdorf, R., Programing PHP, 2002., str. 203.

²⁷ Lerdorf, R., Programing PHP, 2002., str. 205.

```

<?
    $host="localhost";
    $username="ncindric";
    $password="gneogjeor";
    $database="ncindric";

    try{
        $conn =new PDO ("mysql:host=$host; dbname=$database;", $username, $password);
    }
    catch(PDOException $e){
        die("Connection failed:" . $e-> getMessage());
    }
?>

```

Slika 14: Konekcija na bazu podataka

izvor: izrada autora

Na početku rada definirane su varijable \$host, \$username, \$password i \$database i napunjene podacima za prijavu na MySQL server. Za konekciju na bazu podataka korištena je ekstenzija za pristup bazama u PHP-u PDO odnosno PHP Data Objects u koju su unijete prethodno definirane varijable, kako bi se povezali na svoju bazu podataka. PDO pruža pristup podacima na apstrakcijskom sloju, što znači da će neovisno o bazama podataka izdati upite i dohvatiti podatke.²⁸ Ako se server ne uspije povezati na bazu podataka, funkcija PDOException javlja pogrešku nastalu od strane PDO-a.²⁹

Nakon povezivanja na bazu podataka moguće je vršiti osnovne radnje uporabom MySQL-a kao što je unos podataka u bazu, čitanje iz baze podataka, brisanje iz baze podataka i izmjena podataka u bazi podataka. Započeto je unosom podataka u bazu podatka. Da bi unijeli podatke u bazu podataka, potrebno ih je prikupiti s HTML formom, pohraniti ih korištenjem PHP-a, te uporabom SQL-a unijeti u bazu podataka, kao što je prikazano na slici 15.

²⁸PHP Manula, PDO, dostupno na: <http://php.net/manual/en/intro.pdo.php>, pristupljeno(5.4.2017.)

²⁹ PHP Manual, PDO Exception Class, dostupno na: <http://php.net/manual/en/class.pdoexception.php>, (pristupljeno: 5.4.2017.)

```

<?php
include "inc_conn.php";

$sql="insert into Obrada_Gotova (zaposlenik,pacijent,DTP,komentar,datum,vrijeme,vlasnik) VALUES
('".$_POST['zaposlenik']. "','".$_POST['pacijent']. "','",
('".$_POST['DTP']. "','".$_POST['komentar']. "','".$_POST['datum']. "','",
('".$_POST['vrijeme']. "','".$_POST['vlasnik']. "','"));
if($rez=mysqli_query($conn, $sql)){
    echo 3; die();
}
?>

```

Slika 15: Unos podataka u bazu podataka

izvor: izrada autora

Potrebno se je spojiti na bazu podataka da bi upisali te podatke, što je u danom primjeru napravljeno pozivom datoteke 'inc_conn.php' u kojemu se nalazi kod za povezivanje na bazu. Nakon povezivanja na bazu, definira se varijabla \$sql u koju se upisuje SQL upit za unos podataka u bazu podatka : „insert into *naziv tablice (naziv polja u koje se želi upisati podatak)* VALUES (('".\$_POST['naziv polja u HTML formi']. "','"). Nakon definiranja polja i podataka koji se unose u ta polja, koristi se funkcija koja obavlja upit na bazu podataka mysqli_query koja sadrži attribute (\$conn- konekcija na bazu, \$sql – SQL kod), a pohranjena je u varijablu \$rez.

Nakon što su podatci uneseni u bazu podataka, potrebno ih je iščitati iz baze podataka kako bi ih mogli koristiti na dinamičkoj web stranici ili web aplikaciji, kao što je prikazano na slici 16.

```

$sql1="SELECT a.*, b.OIB, mjesto, adresa FROM Obrada a INNER JOIN Pacijenti b ON a.pacijent= b.ime
WHERE a.vlasnik= '$user' AND a.zaposlenik='$user'AND curdate() between a.start and a.kraj AND a.pon='1'";
$rez1=mysqli_query($conn, $sql1);
while ($niz1=mysqli_fetch_array($rez1)){
    $pacijent=$niz1['pacijent'];
    $OIB=$niz1['OIB'];
    $mjesto_stanovanja=$niz1['mjesto'];
    $adresa=$niz1['adresa'];
    $datum= date('Y/m/d');
    $DTP = $niz1['DTP'];
    $vlasnik=$niz1['vlasnik'];
    if($niz1['pon']==1){$pon='ponedjeljak';}else{$pon='';}
    if($niz1['uto']==1){$uto='utorak';} else{$uto='';}
    if($niz1['sri']==1){$sri='srijeda';}else{$sri='';}
    if($niz1['cet']==1){$cet='četvrtak';}else{$cet='';}
    if($niz1['pet']==1){$pet='petak';}else{$pet='';}
}
?>

```

Slika 16: Čitanje iz baze podataka

izvor: izrada autora

Kao i kod unosa podataka u bazu i kod čitanja iz baze podataka potrebna je konekcija na bazu, što je kao i u prethodnom primjeru napravljeno pozivanjem

datoteke `inc_conn.php` koja sadrži kod za konekciju na bazu podataka. Nakon konekcije ponovno se definira varijabla `$sql`, koja je ovog puta ispunjena sa sljedećim SQL upitom: „SQL *a.naziv polja u tablici, b.naziv polja u tablici* FROM *naziv tablice* *a*(oznaka tablice) INNER JOIN *naziv tablice b* ON *a.naziv primarnog ključa=b.naziv stranog ključa* WHERE *uvjeti koje podaci moraju ispuniti*“. Za čitanje definiranih podataka ponovno se koristi varijabla `$rez` u kojoj je pohranjena funkcija `mysqli_query` s atributima `$conn` i `$sql`. Kod čitanja iz baze podataka koristi se i funkcija `mysqli_fetch_array` koja ima atribut `$rez` i u ovom slučaju je pohranjena u varijablu `$niz` koja je uvjet `while` petlje, što označava da se podatci iščitavaju sve dok je SQL upit točan. Podatci učitani iz baze podataka u ovom su slučaju pohranjeni u varijable npr. `$ime=$niz['ime']` iako se mogu koristiti i u obliku `$niz['ime']`.

Brisanje podataka iz baze podataka kao i unos zahtijeva prikupljanje podataka, tj. primarnog ključa tablice korištenjem HTML formi te pohranjivanje tog podatka korištenjem PHP koda, kao što je prikazano na slici 17. Prikupljeni podatci u SQL upitu koriste se za samu operaciju brisanja podataka iz baze. Kao i kod prethodnih operacija s PHP-om i MySQL-om i kod brisanja podataka, potrebna je konekcija na bazu koja je napravljena identično kao i na prethodno opisanim primjerima. Varijabla `$sql` ovoga se puta sastoji od sljedećeg SQL upita: „DELETE FROM *naziv tablice* WHERE *naziv polja primarnog ključa* =“`$_POST['naziv primarnog ključa u formi']`““. Kao i kod prethodnih sql upita i ovaj se upit izvršava uporabom funkcije `mysqli_query` koja sadrži attribute `$conn` i `$sql`, a pohranjena je u varijabli `$rez`.

```
<?php
include "inc_conn.php";
$sql="delete from Obrada_Gotova WHERE komentar='". $_POST['id'] ."'";
$rez=mysqli_query($conn, $sql);
?>
```

Slika 17: Brisanje podataka iz baze podataka

izvor: izrada autora

Podatke koji se nalaze u bazi podataka moguće je i izmijeniti, također njihovim prikupljanjem i pohranjivanjem te konekcijom na bazu kao u već navedenim primjerima (slika 18.). Za razliku od prethodnih primjera, u ovom su primjeru podatci prikupljeni PHP-om koji su pohranjeni u varijabli radi jednostavnijeg daljnjeg korištenja. Varijabla `$sql` u ovom se primjeru sastoji od sljedećeg upita: „UPDATE

naziv tablice SET naziv polja u tablici = varijabla s nazivom naziv polja u formi WHERE naziv primarnog ključa= naziv primarnog ključa u tablici“.

```
<?
include "inc_conn.php";
$id=$_POST['id'];
$text=$_POST['text'];
$column_name=$_POST['column_name'];

$sql="UPDATE Zaposlenici SET ".$column_name."='".$text."' WHERE OIB='".$id."'";
$res=mysqli_query($conn, $sql);
?>
```

Slika 18: Izmjena podataka u bazi podataka

izvor: izrada autora

4.2.AJAX

AJAX je skraćenica za Asinkroni JavaScript + XML. JavaScript je klijent side skriptni jezik koji omogućava dodatne funkcionalnosti web stranica s kojim korisnik može komunicirati po vlastitoj želji. Brendana Eich, zaposlenik Netscape-a, kreirao je 1995. godine JavaScript ili izvorno LiveScript te je prvi put programerima omogućio da sami utječu na to kako će web stranica komunicirati s korisnikom.³⁰ odnosno Prošlo je nevjerojatno malo vremena od samog nastanka JavaScripta do njegove potpuno globalne primjene. JavaScript danas je vjerojatno najvažniji skriptni jezik upravo zbog svoje sposobnosti udruživanja s web preglednikom, zbog čega i nosi nadimak „Language of the Web“. Ovaj je skriptni jezik nastao na nekoliko dobrih, ali i nekoliko loših ideja. Dobre strane JavaScripta su funkcije koje olakšavaju rad, ali i omogućuju manje koda te dinamičnost objekata. Jedna od najvećih mana jest model programiranja temeljen na globalnim varijablama. Okviri i knjižnice JavaScripta u posljednjih nekoliko godina postale su neizmjereno popularne zbog omogućavanja korištenja brojnih tehnologija poput JavaScripta i Ajaxa u tzv. Web 2.0. web aplikacijama. jQuery jedna je od knjižnica JavaScripta čiji je zadatak uložiti mnogo truda kako bi olakšao život web programerima sastavljanjem određenih dijelova preglednika, olakšavajući time ostale prijeko potrebne zadatke. Koristeći knjižničnu građu JavaScript zapravo omogućuje web programerima promjenu konteksta HTML, dokumenta modificiranjem modela koje preglednik stvara otvaranjem web stranice tzv. DOM manipulacije. jQuery ima stvarnu mogućnost pretvaranja nekoliko linija

³⁰ N.C.Zakas, J.McPeak, J.Fawcett, *Profesional Ajax*, Indianapolis, Wiley Publishing, inc., 2006., str.6.

koda JavaScripta u samo jednu liniju koda, pritom nastojeći maknuti redundanciju gdje god je to moguće te normalizirati hibridne preglednike razvijene JavaScriptom u ključnim područjima u kojima se preglednici mogu razlikovati, npr. postavljanje kursora računalnog miša kada se dogodi neka promijena njegovog položaja. Dvije najpoznatije jQuery knjižnice su jQuery UI i jQuery Mobile.³¹ Jesse Jamesa Gerreta predstavila je Ajax kao jednu od metoda jQuerya u članku pod nazivom „Ajax: A New Approach to Web Applications“, u kojemu je navela kako Googleovi projekti poput Google Suggest i Google Maps odbacuju tradicionalnu desktop temeljenu korisničku interakciju i prelaze na web pristup Ajax poziva. Funkcija Ajax-a jest da korisničko sučelje približi što je više moguće idealnome, odnosno da ono bude nevidljivo korisnicima pružajući im opcije koje su im u tome trenutku potrebne, a u suprotnome ih korisnici ne vide. Google i druge velike web tvrtke koriste Ajax kako bi podigle očekivanja korisnika što web zahtjevi korisnika mogu učiniti. Koristeći „prijašnje“ tehnologije u kombinaciji s Ajaxom, mogućnosti aplikacije šire se i izvan njezinih početnih mogućnosti i očekivanja.³² Ajax omogućava dizajniranja i izgradnju web aplikacija tako da postanu interaktivne i vraćaju podatke kao i desktop aplikacije.³³

Jedna od najkorisnijih metoda jQuery-a upravo je \$.ajax() čije su akcije globalne funkcije neovisne o DOM-u. Funkcija \$.ajax() sadrži jedan atribut, odnosno atribut koji sadrži postavke za Ajax pozive; ako je funkcija pozvana bez atributa, stranica će se ažurirati i neće ništa uraditi s rezultatom. Funkcija \$.ajax () može sadržavati sljedeće attribute:

- **data:** predstavlja podatke koji se šalju, mogu biti upisane u obliku znakova (key1=val1&key2=val2) ili u obliku JSON-a ({„key1“:“val1“, „key2“:“val2“})
- **data filter:** omogućuje prefilterizaciju podataka i korisni je za sterilizaciju podataka koji se šalju u skriptu
- **data type:** prikazuje tip podataka npr. „xml“, „html“, „json“, „text“...

³¹ R.York, *Beginning JavaScript and CSS Development with JQuery*, Indianapolis, Wiley Publishing, inc., 2009, dostupno na: https://books.google.hr/books?id=L9otyT4crSQC&printsec=frontcover&dq=jquery&hl=hr&sa=X&ved=0ahUKEwibirjro_TAhUGWhQKHuQAo0Q6AEIOTAE#v=onepage&q=jquery&f=false, str. 19., pristupljeno (6.4.2017.)

³² D.Crane, E.Pascarello, D. James, *Ajax in action*, Manning Publications, inc., 2006., str. 4.

³³ R.Riordan, *Head First Ajax*, O'Reilly Media, Inc., 2008. godine dostupno na: <https://books.google.hr/books?id=sMMX-RWBbDkC&pg=PA9&dq=benefits+of++ajax&hl=hr&sa=X&ved=0ahUKEwieyOrNgpLTAhUJP5oKHemNBdMQ6AEIHAB#v=onepage&q=benefits%20of%20%20ajax&f=false>, str. 2, (pristupljeno: 7.4.2017.)

- **error:** Prikazuje status zahtjeva i prikazuje tekst pogreške
- **success:** ovaj se atribut provodi ako je zahtjev u potpunosti uspješno proveden
- **type:** prikazuje koji tip zahtjeva je poslan POST, GET, DELETE...
- **url:** prikazuje url na koji se zahtjev šalje³⁴.

```
<script>
$(document).ready(function(){
    $('[data-submit]').click(function(){

        $.ajax({
            type: "POST",
            url: "add_medias.php",
            data: $(this).parent().serialize(),
            success: function(response){
                if (response === '3') { alert('Uspješno izvršena obrada!'); window.location = 'medias.php'; }
                else { alert('Ups! Obrada nije dobro izvršena, molimo Vas da pokušate ponovno!'); }
            }
        });
    });
});
</script>
```

Slika 19: primjena \$.ajax() poziva

izvor: izrada autora

Na slici 19. prikazan je primjer uporabe \$.ajax() poziva na php korištenih u aplikaciji „Medias“. Funkcija \$.ajax() sadrži attribute type, url, data i success. Atribut data opisuje tip zahtjeva, odnosno poziva na php u slučaju sa slike 19. radi se o POST zahtjevu. Atribut url prikazuje url adresu na koju se šalju podatci pomoću funkcije \$.ajax(); u prikazanom slučaju radi se o url-u add_medias.php na kojemu se nalazi mySQLi kod koji upisuje podatke u bazu podataka. Atribut data sadrži podatke koji se putem funkcije \$.ajax() šalju na definirani url. U slučaju sa slike 19. data sadržava \$('#medias').serialize() što označava da se na definirani url šalju svi podatci koji su prikupljeni kroz formu koja ima id medias. Posljednji atribut koji u primjeru ima funkcija \$.ajax() je success te koji označava radnju koja će se dogoditi ako se zahtjev uspješno provede. U ovom slučaju radi se o funkciji koja ovisi o odgovoru na stranici add_medias.php na koju se zahtjev šalje. Ako je odgovor na

```
if($rez=mysqli_query($conn, $sql)){ echo 3; die();}
```

³⁴ K.Wald, J. Lengstorf, Pro PHP and JQUERY, dostupno na: <https://books.google.hr/books?id=PyWFCwAAQBAJ&pg=PA77&dq=jquery+ajax+request+on+php&hl=hr&sa=X&ved=0ahUKEwjx3YfsjZLTAhWHISwKHUmmAg0Q6AEIFzAA#v=onepage&q=jquery%20ajax%20request%20on%20php&f=false> , 2016., str.77., (pristupljeno: 7.4.2017.)

točan, odnosno odgovor je identičan 3, u tom se slučaju javlja upozorenje : „Uspješno izvršena obrada!“ te nakon klika na upozorenje ažurira se stranica medias.php. Ako odgovor nije točan, odnosno nije identičan 3, tada se javlja upozorenje: „Ups! Obrada nije dobro izvršena, molimo vas da pokušate ponovno!“

```
<script>
$(document).ready(function(){

    function fetch_data()
    {
        $.ajax({
            type:"POST",
            url: "popis_pacijenata1.php",
            success: function(data) {
                $('#pacijenti').html(data);
            }
        });
    }
    fetch_data ();
});
```

Slika 20: Prikaz podataka

izvor: izrada autora

Osim što se mogu slati podatci za upis u bazu podataka putem funkcije \$.ajax(), mogu se i samo prikazivati podatci bez slanja podataka, kao što je prikazano na slici 20. Funkcija fetch_data pomoću \$.ajax() prikazuje ažurirane podatke na stranici bez potrebe za ažuriranjem cijele stranice. U ovom slučaju metoda \$.ajax() sadrži attribute type, url i success. Atribut type označava tip zahtjeva koji je poslan, a u ovom slučaju radi se o POST tipu zahtjeva. Atribut url prikazuje url na koji se šalje zahtjev, što je u ovom slučaju „aktivni_pacijenti1.php“, te atribut success prikazuje funkciju dana koja označava da se prikazuju podatci koji se nalaze u tablici koja ima id aktivni_pacijenti.

```

function edit_data(id,text, column_name){
    $.ajax({
        type:"POST",
        url:"edit_pacijenti.php",
        data:{id:id,text:text,column_name:column_name},
        success:function(data){
            alert(data)
        }
    });
};
$(document).on('blur','.ime', function(){
    var id=$(this).data("id1");
    var ime=$(this).text();
    edit_data(id, ime, "ime" );
    alert(ime);
});
$(document).on('blur','.OIB', function(){
    var id=$(this).data("id2");
    var OIB=$(this).text();
    edit_data(id, OIB, "OIB" );
    alert(OIB);
});
$(document).on('blur','.lijecnik', function(){
    var id=$(this).data("id3");
    var lijecnik=$(this).text();
    edit_data(id, lijecnik, "lijecnik" );
    alert(lijecnik);
});
$(document).on('blur','.mjesto', function(){
    var id=$(this).data("id4");
    var mjesto=$(this).text();
    edit_data(id, mjesto, "mjesto" );
    alert(mjesto);
});

```

Slika 21: \$.ajax() Izmjena podataka

izvor: izrada autora

Pomoću \$.ajax() možemo mijenjati podatke kao što je prikazano na slici 21. Prvo je postavljena funkcija edit_data gdje su kao parametri navedeni podatci koji se šalju u dalju skriptu i potrebni su za izmjenu podataka u bazi podataka, a u ovom slučaju su to id, text i column_name. Funkcija edit_data sastoji se od metode \$.ajax() koja sadrži atribut type, url, data i success. I u ovom primjeru tip zahtjeva koji se šalje putem metode \$.ajax () je POST, url na koji se podaci šalju je „edit_aktivni.php“, data sadrži podatke koji su navedeni kao parametri funkcije edit_data zapisane u json obliku. Atribut success prikazuje upozorenje s podatkom koji je upisan ako se zahtjev dobro provede. Nakon definiranja edit_data pomoću metode \$(document).on('blur', 'id_polja, function()') omogućena je izmjena podataka unutar tablice. Za svako polje u kojemu želimo mijenjati podatke potrebno je unutar funkcije definirati varijablu id koja će sadržavati id polja u kojemu želimo mijenjati podatke i varijablu s nazivom tog polja koju definiramo kao \$(this).text();. nakon definiranja varijabli potrebno je pozvati

prethodno definiranu funkciju `edit_data` koja će sadržavati varijable `id`, naziv polja i naziv stupca u bazi podataka u kojemu se mijenja podatak.

Kod tradicionalnih web stranica i aplikacija svaki put kada korisnik klikne na nešto odnosno komunicira sa serverom, preglednik šalje zahtjev korisnika serveru nakon čega server odgovara ažuriranjem cijele stranice, što dovodi do trošenja vremena korisnika. Korištenjem Ajaxa preglednik pita server samo za one podatke koji su korisniku potrebni, te server odgovara ažuriranjem samo jednog dijela stranice, što u konačnici dovodi do manjeg čekanja traženih podataka. Ajax između ostalog omogućuje da preglednik šalje nekoliko upita serveru odjednom, no korisnik toga ponekad nije ni svjestan jer može odmah nastaviti s radom, a ne čekati da se stranica ažurira. Ovisno o količini podataka, Ajax omogućuje brži odgovor servera s traženim podacima, no ako se radi o velikoj količini podataka, moguće je da stranica napisana u Ajax-u ima sporije odgovore od tradicionalnih web aplikacija i stranica. Pomoću Ajax-a „promet“ između preglednika i servera je olakšan, no kako Ajax omogućuje istovremeno slanje više zahtjeva i odgovora između preglednika i servera, može se stvoriti zagušenje tog prometa. Ajax omogućuje korisniku nastavak rada u aplikaciji neovisno o tom jesu li podaci koje je korisnik tražio ažurirani ili su u procesu ažuriranja, odnosno korisnik ne mora čekati završetak odgovora servera kako bi mogao nastaviti s radom na aplikaciji.³⁵

³⁵ R.Riordan, *Head First Ajax*, O'Reilly Media, Inc., 2008. godine dostupno na: <https://books.google.hr/books?id=sMMX-RWBbDkC&pg=PA9&dq=benefits+of++ajax&hl=hr&sa=X&ved=0ahUKEwieyOrNgpLTAhUJP5oKHemNBdMQ6AEIHAB#v=onepage&q=benefits%20of%20%20ajax&f=false>, str. 10., (pristupljeno: 7.4.2017.)

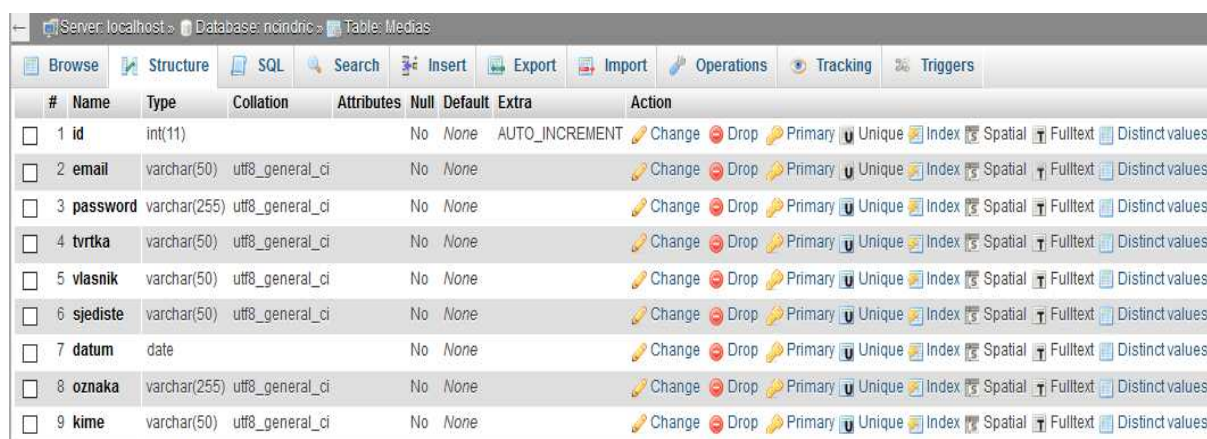
5. Primjer aplikacije

Web aplikacija „Medias“ aplikacija je koja je namijenjena vlasnicima ustanova za pružanje kućne njege. Vlasnici takvih ustanova putem aplikacije „Medias“ mogu na jednom mjestu unositi svoje pacijente i zaposlenike, definirati obrade pojedinih pacijenata koje vrše pojedini zaposlenici, pratiti rad zaposlenika, pratiti pacijente koje sami moraju obraditi, te na temelju izvještaja pratiti napredak pacijenata, ali i definirati plaće zaposlenika.

Aplikacija je nastala na temelju evolucijskog modela razvoja softvera. Na temelju opisa problema koje su predočile vlasnice ustanova za pružanje kućne njege, izrađena je prva verzija aplikacije „Medias“. Nakon sugestija vlasnica ustanova o potrebi za povećanjem mogućih pacijenata, mogućnosti brisanja i izmjena podataka u tablicama izrađena je nova verzija aplikacije koja je zadovoljila potrebe vlasnica ustanova. Aplikacija je izrađena koristeći web editor Codeanywhere te phpMyAdmin bazu podataka. Osim HTML kao osnovnog jezika za izradu web stranica i aplikacija, pozadinska strana aplikacije izrađena je koristeći server-side skriptni jezik PHP uz uporabu MySQL baze podataka, dok je za dizajn aplikacije korišten okvir Bootstrap koji je omogućio prilagodbu aplikacije mobilnim uređajima i CSS. Aplikacije se sastoji od 15 dijelova koji će biti razrađeni u nastavku, a aplikaciji je moguće pristupiti na adresi: <http://studlamp.unipu.hr/~ncindric/medias/index.php>.

5.1. Baza podataka aplikacije „Medias“

Baza podataka, izrađena za potrebe aplikacije, nastala je korištenjem online softvera MyPHP Admin i sastoji se od tablica „Medias“, „Pacijenti“, „Zaposlenici“, „DTP“, „Obrada“ i „Obrada_Gotova“. Tablica „Medias“ (slika 22.) tablica je u bazi podataka u koju se prilikom registracije u sustav pohranjuju svi podaci o vlasnicima ustanova za pružanje kućne njege. Sastoji se od sljedećih polja: id, koje je auto-increment, tj. automatski se definira brojem maksimalne veličine 11 znakova; email, tipa znakova veličine 50 znakova; password, tipa znakova veličine 255 znakova; tvrtka, tipa znakova veličine 50 znakova, vlasnik, tipa znakova veličine 50 znakova; sjedište, tipa znakova veličine 50 znakova; datum tipa datum; oznaka, tipa znakova veličine 255 znakova; kime tipa znakova veličine 50 znakova koje je jedinstveno polje, što znači da se unos podataka u tablicu „Medias“ neće izvršiti ako već postoji takvo korisničko ime.



#	Name	Type	Collation	Attributes	Null	Default	Extra	Action
<input type="checkbox"/>	1 id	int(11)			No	None	AUTO_INCREMENT	Change Drop Primary Unique Index Spatial Fulltext Distinct values
<input type="checkbox"/>	2 email	varchar(50)	utf8_general_ci		No	None		Change Drop Primary Unique Index Spatial Fulltext Distinct values
<input type="checkbox"/>	3 password	varchar(255)	utf8_general_ci		No	None		Change Drop Primary Unique Index Spatial Fulltext Distinct values
<input type="checkbox"/>	4 tvrtka	varchar(50)	utf8_general_ci		No	None		Change Drop Primary Unique Index Spatial Fulltext Distinct values
<input type="checkbox"/>	5 vlasnik	varchar(50)	utf8_general_ci		No	None		Change Drop Primary Unique Index Spatial Fulltext Distinct values
<input type="checkbox"/>	6 sjedište	varchar(50)	utf8_general_ci		No	None		Change Drop Primary Unique Index Spatial Fulltext Distinct values
<input type="checkbox"/>	7 datum	date			No	None		Change Drop Primary Unique Index Spatial Fulltext Distinct values
<input type="checkbox"/>	8 oznaka	varchar(255)	utf8_general_ci		No	None		Change Drop Primary Unique Index Spatial Fulltext Distinct values
<input type="checkbox"/>	9 kime	varchar(50)	utf8_general_ci		No	None		Change Drop Primary Unique Index Spatial Fulltext Distinct values

Slika 22: Prikaz strukture tablice „Medias“

izvor: izrada autora

Tablica „Pacijenti“ (slika 23.), koja se nalazi u bazi podataka korištenoj za potrebe izrade ovoga diplomskog rada, pohranjuje podatke koji se unose na stranici unos pacijenata i podatke koji se mijenjaju na stranici popis pacijenata. Tablica „Pacijenti“ sastoji se od sedam polja, a to su: OIB, mjesto, spol, liječnik, vlasnik, adresa, ime. Polje OIB, koje je jedinstveno, omogućava unos brojeva do 11 mjesta, što znači da se neće izvršiti upis u bazu pacijenta s OIB-om koji se već nalazi u bazi podataka. Polje mjesto omogućava unos do 50 znakova, polje spol do 2 znaka, polja spol, liječnik, vlasnik, adresa i ime su polja koja omogućavaju unos do 50 znakova.

#	Name	Type	Collation	Attributes	Null	Default	Extra	Action
<input type="checkbox"/>	1 OIB	int(11)			No	None		Change Drop Primary Unique Index Spatial Fulltext Distinct values
<input type="checkbox"/>	2 mjesto	varchar(50)	utf8_general_ci		No	None		Change Drop Primary Unique Index Spatial Fulltext Distinct values
<input type="checkbox"/>	3 spol	varchar(2)	utf8_general_ci		No	None		Change Drop Primary Unique Index Spatial Fulltext Distinct values
<input type="checkbox"/>	4 lijecnik	varchar(50)	utf8_general_ci		No	None		Change Drop Primary Unique Index Spatial Fulltext Distinct values
<input type="checkbox"/>	5 vlasnik	varchar(50)	utf8_general_ci		No	None		Change Drop Primary Unique Index Spatial Fulltext Distinct values
<input type="checkbox"/>	6 adresa	varchar(50)	utf8_general_ci		No	None		Change Drop Primary Unique Index Spatial Fulltext Distinct values
<input type="checkbox"/>	7 ime	varchar(50)	utf8_general_ci		No	None		Change Drop Primary Unique Index Spatial Fulltext Distinct values

Slika 23: Prikaz strukture tablice "Pacijenti"

izvor: izrada autora

Tablica „Zaposlenici“ (slika 24.) sastoji se od 10 polja koja se popunjavaju korištenjem stranice Unos zaposlenika ili mijenjaju korištenjem stranice Popis zaposlenika. Polja koja sadrži ova tablica su: id_zaposlenika, ime, spol, broj_prijave, korisnicko_ime, datum, mjesto Stanovanja, mjesto rođenja, vlasnik, prezime. Polje id_zaposlenika omogućava unos do 11 brojeva koje se popunjava automatski, tj. auto_increment, polje ime omogućava unos do 50 znakova, polje spol do 2 znaka, polje broj_prijave do 300 znakova, polje korisnicko_ime do 50 znakova, polje datum označava datum rođenja zaposlenika i omogućava unos datuma te polja mjesto Stanovanja, mjesto rođenja, vlasnik i prezime do 50 znakova.

#	Name	Type	Collation	Attributes	Null	Default	Extra	Action
<input type="checkbox"/>	1 id_zaposlenika	int(11)			No	None	AUTO_INCREMENT	Change Drop Primary Unique More
<input type="checkbox"/>	2 ime	varchar(50)	utf8_general_ci		No	None		Change Drop Primary Unique More
<input type="checkbox"/>	3 spol	varchar(2)	utf8_general_ci		No	None		Change Drop Primary Unique More
<input type="checkbox"/>	4 broj_prijave	varchar(300)	utf8_general_ci		No	None		Change Drop Primary Unique More
<input type="checkbox"/>	5 korisnicko_ime	varchar(50)	utf8_general_ci		No	None		Change Drop Primary Unique More
<input type="checkbox"/>	6 datum	date			No	None		Change Drop Primary Unique More
<input type="checkbox"/>	7 mjesto Stanovanja	varchar(50)	utf8_general_ci		No	None		Change Drop Primary Unique More
<input type="checkbox"/>	8 mjesto rođenja	varchar(50)	utf8_general_ci		No	None		Change Drop Primary Unique More
<input type="checkbox"/>	9 vlasnik	varchar(50)	utf8_general_ci		No	None		Change Drop Primary Unique More
<input type="checkbox"/>	10 prezime	varchar(50)	utf8_general_ci		No	None		Change Drop Primary Unique More

Slika 24:Prikaz strukture tablice "Zaposlenici"

izvor: izrada autora

Tablica „DTP“ (slika 25.) sadrži trajne podatke, tj. u ovu tablicu ne upisuju se podatci, već se samo iz nje iščitavaju. Tablica sadrži podatke o postupcima obrade pacijenta koje je definirao HZZ i vrijede za sve ustanove jednako. Tablica se sastoji od 4 polja i to: id_dtp, DTP, cijena i vrijeme. Polje id_dtp je auto_increment koje omogućava unos do 11 brojeva, polje DTP označava postupak i unos do 100

znakova, polje cijena označava jediničnu cijenu pojedinog postupka i unos do 11 brojeva, polje vrijeme označava vrijeme trajanja pojedinog postupka obrade pacijenta i unos do 11 brojeva.

#	Name	Type	Collation	Attributes	Null	Default	Extra	Action
<input type="checkbox"/>	1 id_dtp	int(11)			No	None	AUTO_INCREMENT	Change Drop Primary Unique Index Spatial Fulltext More
<input type="checkbox"/>	2 DTP	varchar(100)	utf8_general_ci		No	None		Change Drop Primary Unique Index Spatial Fulltext More
<input type="checkbox"/>	3 cijena	int(11)			No	None		Change Drop Primary Unique Index Spatial Fulltext More
<input type="checkbox"/>	4 vrijeme	int(11)			No	None		Change Drop Primary Unique Index Spatial Fulltext More

Slika 25: Prikaz strukture tablice „DTP“

izvor: izrada autora

Tablica „Obrada“ (slika 26.) sadrži podatke koje definira vlasnik ustanove na stranici Obrada te podatke koji su izmijenjeni na stranici Aktivni pacijenti. Tablica sadrži 15 polja: id_obrade, pacijent, DTP, start, kraj, pon, uto, sri, cet, pet, zaposlenik, zbroj, ukupno_vrijeme, ukupna_cijena, vlasnik. Polje id_obrade jedinstveno je polje koje omogućava unos do 11 brojeva; polja pacijent i DTP unos do 50 znakova; polja start i kraj označavaju datum početka i kraja obrade pacijenta i omogućavaju unos datuma; polja pon, uto, sri, cet, pet polja su koja označavaju dan ponavljanja obrade i unos do 11 brojeva; polje zaposlenik unos do 50 znakova; polja ukupno_vrijeme i ukupna_cijena unos do 11 brojeva i polje vlasnik do 50 znakova.

#	Name	Type	Collation	Attributes	Null	Default	Extra	Action
<input type="checkbox"/>	1 id_obrade	int(11)			No	None		Change Drop Primary Unique Index Spatial Fulltext Distinct values
<input type="checkbox"/>	2 pacijent	varchar(50)	utf8_general_ci		No	None		Change Drop Primary Unique Index Spatial Fulltext Distinct values
<input type="checkbox"/>	3 DTP	varchar(50)	utf8_general_ci		No	None		Change Drop Primary Unique Index Spatial Fulltext Distinct values
<input type="checkbox"/>	4 start	date			No	None		Change Drop Primary Unique Index Spatial Fulltext Distinct values
<input type="checkbox"/>	5 kraj	date			No	None		Change Drop Primary Unique Index Spatial Fulltext Distinct values
<input type="checkbox"/>	6 pon	int(11)			No	None		Change Drop Primary Unique Index Spatial Fulltext Distinct values
<input type="checkbox"/>	7 uto	int(11)			No	None		Change Drop Primary Unique Index Spatial Fulltext Distinct values
<input type="checkbox"/>	8 sri	int(11)			No	None		Change Drop Primary Unique Index Spatial Fulltext Distinct values
<input type="checkbox"/>	9 cet	int(11)			No	None		Change Drop Primary Unique Index Spatial Fulltext Distinct values
<input type="checkbox"/>	10 pet	int(11)			No	None		Change Drop Primary Unique Index Spatial Fulltext Distinct values
<input type="checkbox"/>	11 zaposlenik	varchar(50)	utf8_general_ci		No	None		Change Drop Primary Unique Index Spatial Fulltext Distinct values
<input type="checkbox"/>	12 zbroj	int(11)			No	None		Change Drop Primary Unique Index Spatial Fulltext Distinct values
<input type="checkbox"/>	13 ukupno_vrijeme	int(11)			No	None		Change Drop Primary Unique Index Spatial Fulltext Distinct values
<input type="checkbox"/>	14 ukupna_cijena	int(11)			No	None		Change Drop Primary Unique Index Spatial Fulltext Distinct values
<input type="checkbox"/>	15 vlasnik	varchar(50)	utf8_general_ci		No	None		Change Drop Primary Unique Index Spatial Fulltext Distinct values

Slika 26: Prikaz strukture tablice „Obrada“

izvor: izrada autora

U tablicu „Obrada_Gotova“ (slika 27.) pohranjuju se podatci nakon izvršenja obrade, što zaposlenici i vlasnici evidentiraju putem stranica Medias i MediasK. Tablica se sastoji od osam polja: id_og, zaposlenik, pacijent, DTP, komentar, datum, vrijeme i vlasnik. Polje id_og je auto_increment koje omogućava unos do 11 brojeva; polja zaposlenik, pacijent i DTP unos do 50 znakova; u polje komentar može se unositi neograničen broj znakova, a u polja datum i vrijeme unose se podatci za termin obrade; polje vlasnik omogućava unos do 50 znakova.

#	Name	Type	Collation	Attributes	Null	Default	Extra	Action
<input type="checkbox"/>	1 id_og	int(11)			No	None	AUTO_INCREMENT	Change Drop Primary Unique Index Spatial Fulltext Distinct values
<input type="checkbox"/>	2 zaposlenik	varchar(50)	utf8_general_ci		No	None		Change Drop Primary Unique Index Spatial Fulltext Distinct values
<input type="checkbox"/>	3 pacijent	varchar(50)	utf8_general_ci		No	None		Change Drop Primary Unique Index Spatial Fulltext Distinct values
<input type="checkbox"/>	4 DTP	varchar(50)	utf8_general_ci		No	None		Change Drop Primary Unique Index Spatial Fulltext Distinct values
<input type="checkbox"/>	5 komentar	text	utf8_general_ci		No	None		Change Drop Primary Unique Index Spatial Fulltext Distinct values
<input type="checkbox"/>	6 datum	date			No	None		Change Drop Primary Unique Index Spatial Fulltext Distinct values
<input type="checkbox"/>	7 vrijeme	time			No	None		Change Drop Primary Unique Index Spatial Fulltext Distinct values
<input type="checkbox"/>	8 vlasnik	varchar(50)	utf8_general_ci		No	None		Change Drop Primary Unique Index Spatial Fulltext Distinct values

Slika 27: Prikaz strukture tablice „Obrada_Gotova“

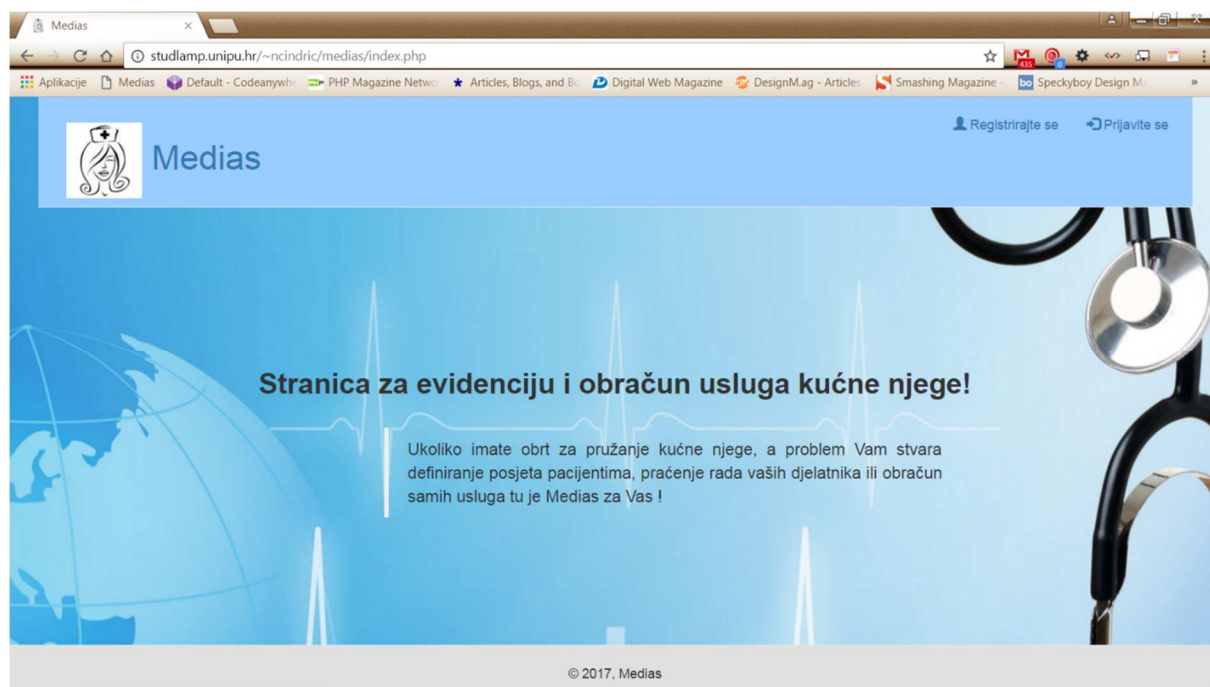
izvor: izrada autora

5.2. Dinamička web aplikacija „Medias“

Nakon upoznavanja s tablicama u bazi podataka za potrebe dinamičke web aplikacije „Medias“, bit će prikazane i opisane stranice od kojih se sastoji aplikacija.

5.2.1. Početna stranica

Početna stranica „Medias“ sastoji se od zaglavlja kojega dijeli sa stranicama registracije i prijave. Zaglavlje se sastoji od loga i naziva aplikacije u lijevom kutu te linkova za registraciju i prijavu u desnom kutu. Početna stranica osim zaglavlja ima i glavni dio gdje se nalazi pozadinska slika aplikacije i opis aplikacije. Na dnu početne stranice, kao i na dnu svake stranice ove aplikacije, nalazi se podnožje u kojemu stoji godina izrade i naziv aplikacije. Početna stranica aplikacije „Medias“ vidi se na slici 28.



Slika 28: Početna stranica Medias

izvor: izrada autora

5.2.2. Registracija

Ako vlasnici ustanove za pružanje kućne njege nisu registrirani u sustav, to moraju uraditi na stranici registracija koja se sastoji od zaglavlja i podnožja kao i kod početne stranice te forme koja se sastoji od osam polja od kojih su četiri tekstualna, jedan datum, jedan e-mail i dva password polja. Posljednja tri polja označena su kao obavezna jer se putem njih vlasnici ustanova prijavljuju u sustav. Polje za korisničko ime mora biti jedinstveno dok su polja za lozinku i oznaku šifrirana. Pritiskom na gumb registriraj uporabom jQuery Ajax-a i PHP-a podatci uneseni u formu upisuju se u bazu podataka, te se korisniku omogućava prijava u sustav. Stranica registracije vidi se na slici 29.

Medias

Registrirajte se Prijavite se

Molimo Vas da svoju tvrtku prijavite u sustav.

Naziv tvrtke:
Medicol d.o.o.

Vlasnik tvrtke:
Marija Anić

Sjedište tvrtke:
Pula

Datum osnivanja:
03.03.2005.

Email:
medicol@gmail.com

Korisničko ime:
Marija_Anić

Password:

Oznaka:

Registriraj

Ukoliko imate registriranu tvrtku, molimo da se prijavite. Hvala!

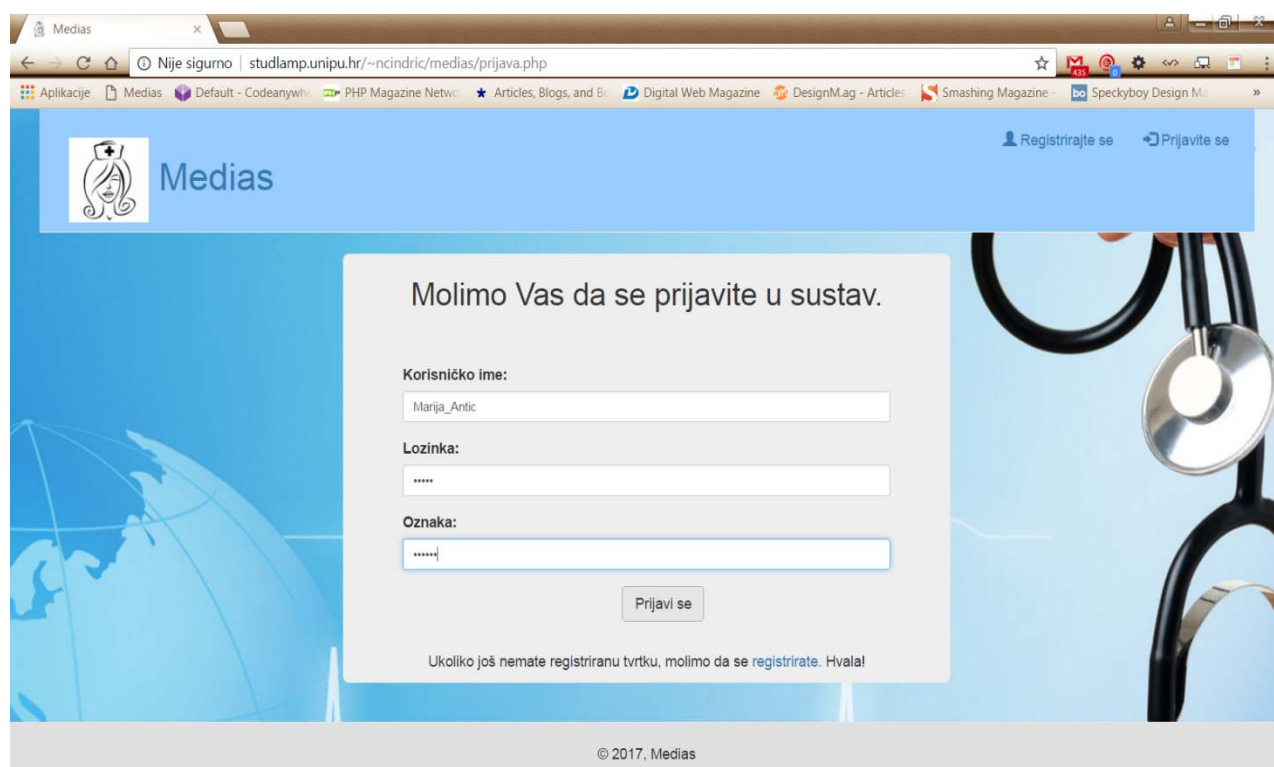
© 2017, Medias

Slika 29: „Medias“ registracija

izvor: izrada autora

5.2.3. Prijava

Kao i prethodne dvije stranice aplikacije i stranica za prijavu sastoji se od zaglavlja i podnožja. U središnjem dijelu stranice nalazi se forma od 4 polja od kojih su dva password polja i jedno tekstualno polje, te omogućavaju prijavu u sustav. Nakon registracije vlasnici ustanova prijavljuju se u sustav unošenjem odabranoga korisničkog imena, lozinke i oznake. Na ovoj stranici aplikacije u sustav prijavljuju se i zaposlenici ustanova za pružanje kućne njege unošenjem korisničkog imena i lozinke koje unutar same aplikacije definira vlasnik. Sustav aplikacije za vlasnike i zaposlenike je različit. Stranica prijave vidi se na slici 30.

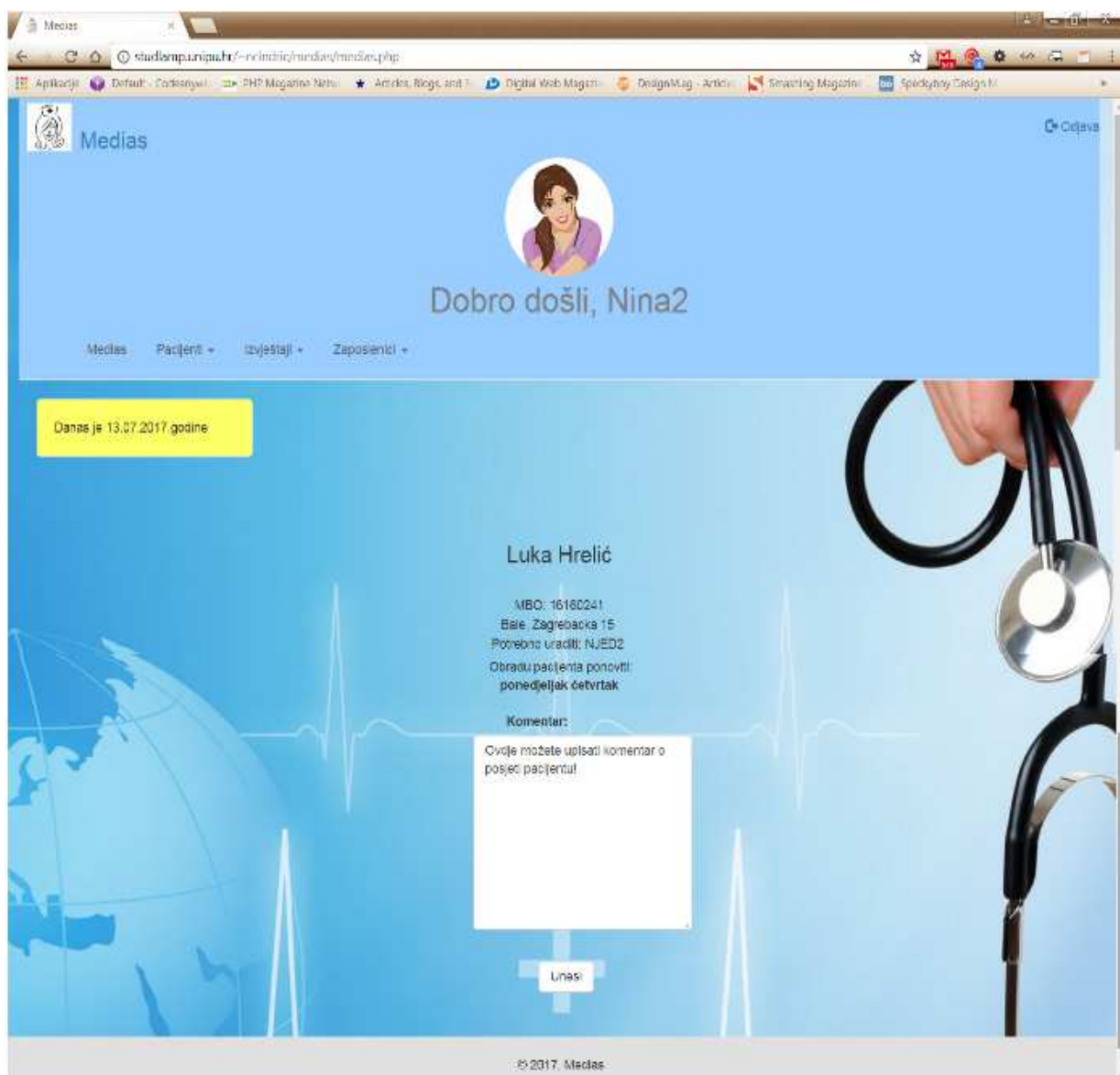


Slika 30: „Medias“ prijava

izvor: izrada autora

5.2.4. Medias

Nakon prijave u sustav vlasnici ustanove dolaze na stranicu medias koja se sastoji od zaglavlja koje se razlikuje od zaglavlja na prethodnim stranicama, podnožja te glavnog djela. Zaglavlje stranice kao i sva zaglavlja nakon prijave u sustav sastoje se od loga i naziva aplikacije u lijevom kutu zaglavlja, dok se u desnom kutu nalazi link za odjavu iz sustava. U sredini zaglavlja nalazi se slika profila korisnika te naziv „Dobro došli, ime korisnika“. Ispod naslova i slike profila nalazi se navigacijska traka s linkovima za ostale stranice aplikacije, a glavne stranice su pacijenti, izvještaji i zaposlenici. Glavni dio ove stranice sastoji se od pravokutnika u lijevom gornjem kutu u kojemu se nalaze vrijeme i sat. U sredini glavnog dijela nalazi se pravokutnik u kojemu su koristeći se PHP-om iz baze podataka izvučeni podatci o imenu, OIB-u, mjestu i adresi, njezi koju pacijent treba te danu u tjednu kada pacijenta treba obraditi. Ispod ispisanih podataka o pacijentima, koji su aktivni za obradu, nalazi se polje u kojemu vlasnici mogu upisati komentar o obradi pojedinog pacijenta, a koje se koristeći jQuery Ajax i PHP s prethodno navedenim podacima te vremenom i datumom šalju u bazu podataka, a vlasniku su vidljivi na stranici praćenje rada. Stranica medias vidi se na slici 31.



Slika 31: Medias

izvor: izrada autora

5.2.5. Unos pacijenata

Link pacijenti na navigacijskoj traci sastoji se od sljedećih linkova: unos pacijenata, obrada, popis pacijenata i popis aktivnih pacijenata. Klikom na link unos pacijenata dolazi se na stranicu koja omogućava unos novih pacijenata u bazu podataka. Ova stranica sastoji se od zaglavlja i podnožja istog kao i stranica medias te glavnog dijela stranice. U glavnom dijelu stranice nalazi se forma od 6 polja, od kojih su četiri tekstualna polja, jedno brojučano polje i jedno polje padajućeg izbornika. Klikom na gumb Unesi koristeći se jQuery Ajax-om i PHP-om uneseni podaci pohranjuju se u bazu podataka. Ako su podaci uneseni u bazu nakon klika, na gumb unesi javlja se upozorenje: „Uspješno dodan novi pacijent!“ Ako podatci nisu uneseni u bazu podataka, javlja se upozorenje: „Ups! Pacijent nije dodan, molimo Vas da pokušate ponovno!“. Stranica unosa pacijenata vidi se na slici 32.

Medias

studlamp.unipu.hr/~/ncindric/medias/pacijenti.php

studlamp.unipu.hr navodi sljedeće:
Uspješno dodan novi pacijent!

U redu

Dobro došli, Nina2

Medias Pacijenti Izveštaji Zaposlenici

Ovde možete unjeti podatke o pacijentu!

Ime i prezime:
Monika Rević

OIB:
5463728293

Spol:
Ž

Liječnik:
Franjo Kopi

Mjesto stanovanja:
Pula

Adresa:
Zagrebačka 18

Insert

© 2017, Medias

Slika 32: Unos pacijenata

izvor: izrada autora

5.2.6. Obrada

Vlasnici ustanova za pružanje kućne njege kroz aplikaciju mogu definirati podjelu pacijenata pojedinim zaposlenicima putem stranice obrada. Stranica obrada kao i prethodna stranica sastoji se od zaglavlja i podnožja te glavnog dijela u kojemu se nalazi forma s jedanaest polja od kojih su četiri polja padajućeg izbornika napunjena podacima iz baze podataka, dva polja datuma te pet checkbox polja koja označavaju radne dane. Pritiskom na tipku Unesi, podatci se koristeći jQuery Ajax i PHP unose u bazu podataka. Ako se podatci unesu u bazu podataka, javlja se upozorenje: „Uspješno definirana obrada pacijenta!“ Ako podatci nisu uneseni u bazu podataka, javlja se upozorenje: „Ups! Obrada nije dobro definirana, molimo Vas da pokušate ponovno!“ Stranica obrada vidi se na slici 33.

Medias

studlamp.unipu.hr navodi sljedeće:
Uspješno definirana obrada pacijenta!

U redu

Dobro došli, Nina2

Medias Pacijenti Izveštaji Zaposlenici

Unos pacijenta
Obrada
Popis
Aktivni pacijenti

Ovdje definirajte parametre obrade pojedinog pacijenta!

Zaposlenik:
Petar

Vlasnik:
Nina2

Pacijent:
Helena Cindrić

DTP postupak:
NJED1

Datum početka:
09.03.2017.

Datum kraja:
07.04.2017.

☐ Ponedjeljak ☒ Utorak ☐ Srijeda ☐ Četvrtak ☒ Petak

Insert

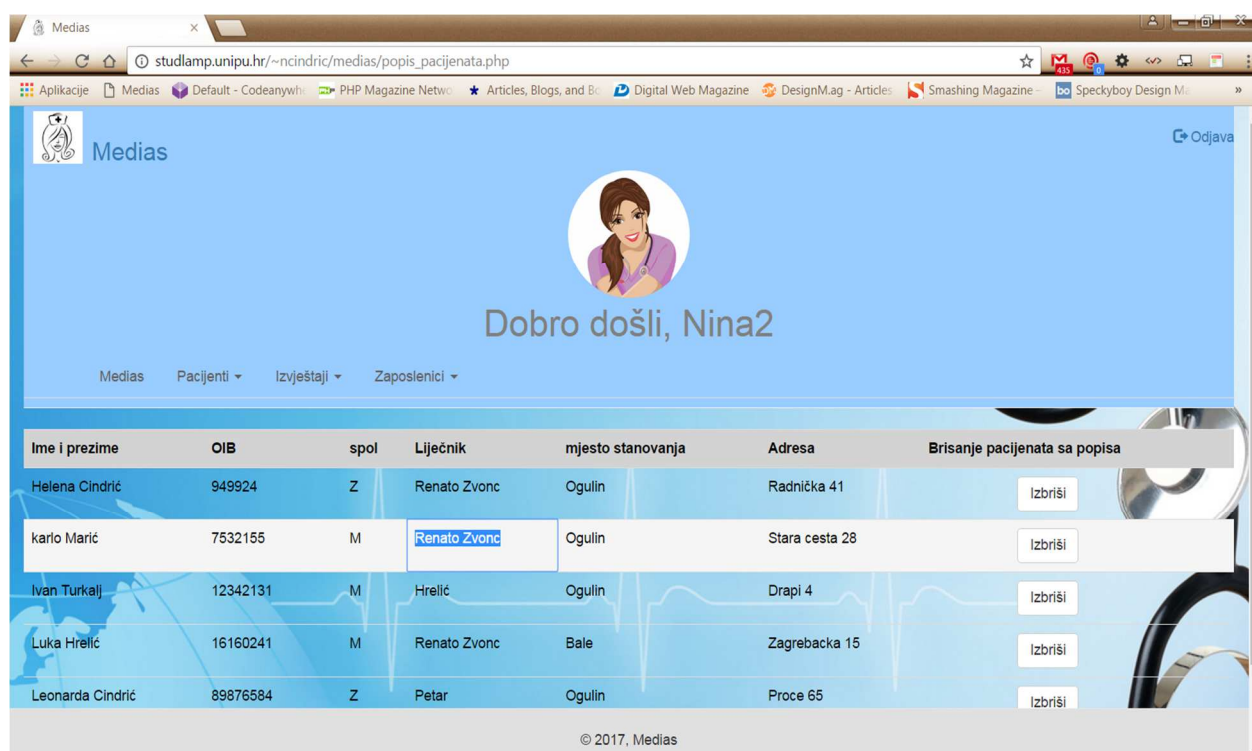
© 2017, Medias

Slika 33: “Medias” obrada

izvor: izrada autora

5.2.7. Popis pacijenata

Stranica popis pacijenata sastoji se od zaglavlja, podnožja te interaktivne tablice koja prikazuje podatke o imenu i prezimenu pacijenta, OIB-u, spolu, liječniku, mjestu stanovanja i adresi pojedinog pacijenta vlasnika ustanove. Tablica omogućava brisanje pojedinog pacijenta iz baze podataka nakon potvrde o brisanju koje se javlja kao upozorenje, a nakon brisanja podataka iz baze podataka, tablica se automatski ažurira. Osim brisanja podataka iz baze podataka, tablica omogućava vlasnicima ustanove da promijene pojedine podatke o pacijentu samim klikom na pojedinu ćeliju tablice koju žele promijeniti. Stranica popisa pacijenata prikazana je na slici 34.



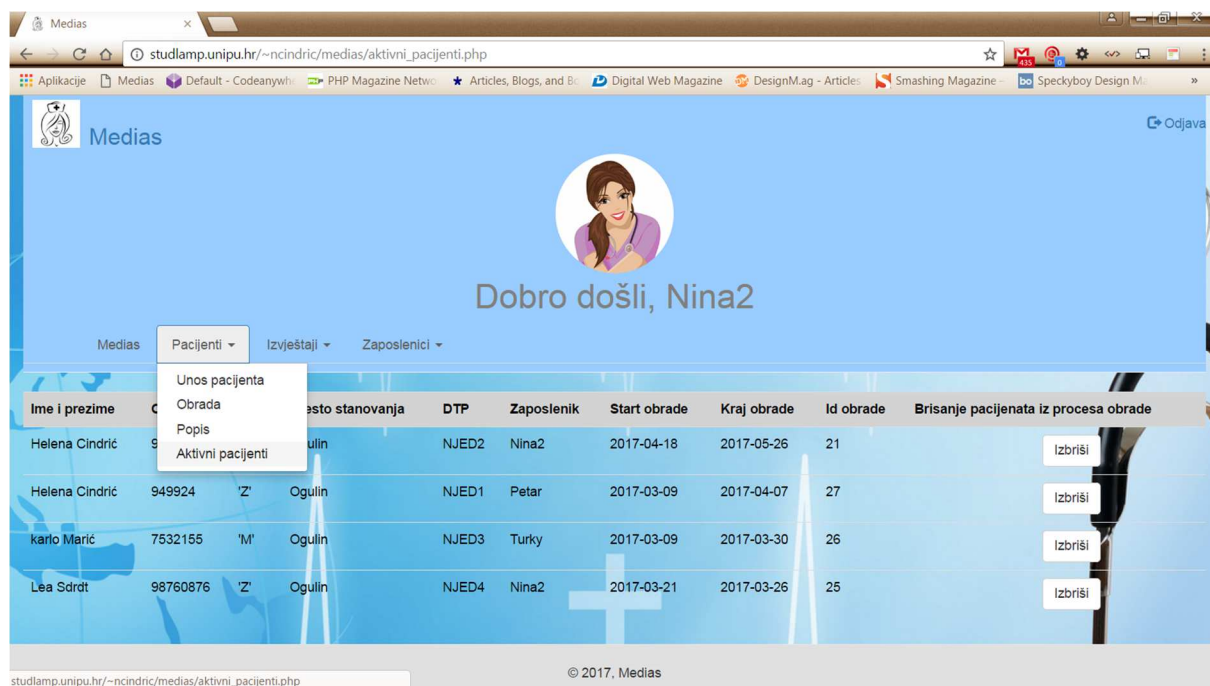
Ime i prezime	OIB	spol	Liječnik	mjesto stanovanja	Adresa	Brisanje pacijenata sa popisa
Helena Cindrić	949924	Z	Renato Zvonc	Ogulin	Radnička 41	Izbriši
Karlo Marić	7532155	M	Renato Zvonc	Ogulin	Stara cesta 28	Izbriši
Ivan Turkalj	12342131	M	Hrelić	Ogulin	Drapl 4	Izbriši
Luka Hrelić	16160241	M	Renato Zvonc	Bale	Zagrebacka 15	Izbriši
Leonarda Cindrić	89876584	Z	Petar	Ogulin	Proce 65	Izbriši

Slika 34: „Medias“ popis pacijenata

izvor: izrada autora

5.2.8. Popis aktivnih pacijenata

Vlasnici ustanova za pružanje kućne njege na stranici popis aktivnih pacijenata mogu pristupiti podacima vezanim uz pacijente čiji je datum početka obrade prošao, a datum kraja obrade slijedi. Stranica popis aktivnih pacijenata istoga je sastava kao i prethodna stranica, no prikazuje podatke iz tablica „Pacijenti“ i „Obrada“, pa su tako vidljivi podatci o imenu i prezimenu, OIB-u, mjestu stanovanja, DTP-u, zaposleniku koji je zadužen za pojedinog pacijenta, datumu početka i kraja obrade te identifikacijskom broju obrade. Kao i prethodna tablica i ova je tablica interaktivna, te omogućuje vlasnicima ustanova brisanje pacijenata iz baze podataka uz automatsko ažuriranje tablice te izmjenu podataka u tablici. U ovoj tablici mogu se mijenjati samo podatci vezani uz obradu pacijenta, a to su DTP, zaposlenik te datum početka i kraja obrade. Stranica aktivni pacijenti prikazana je na slici 35.

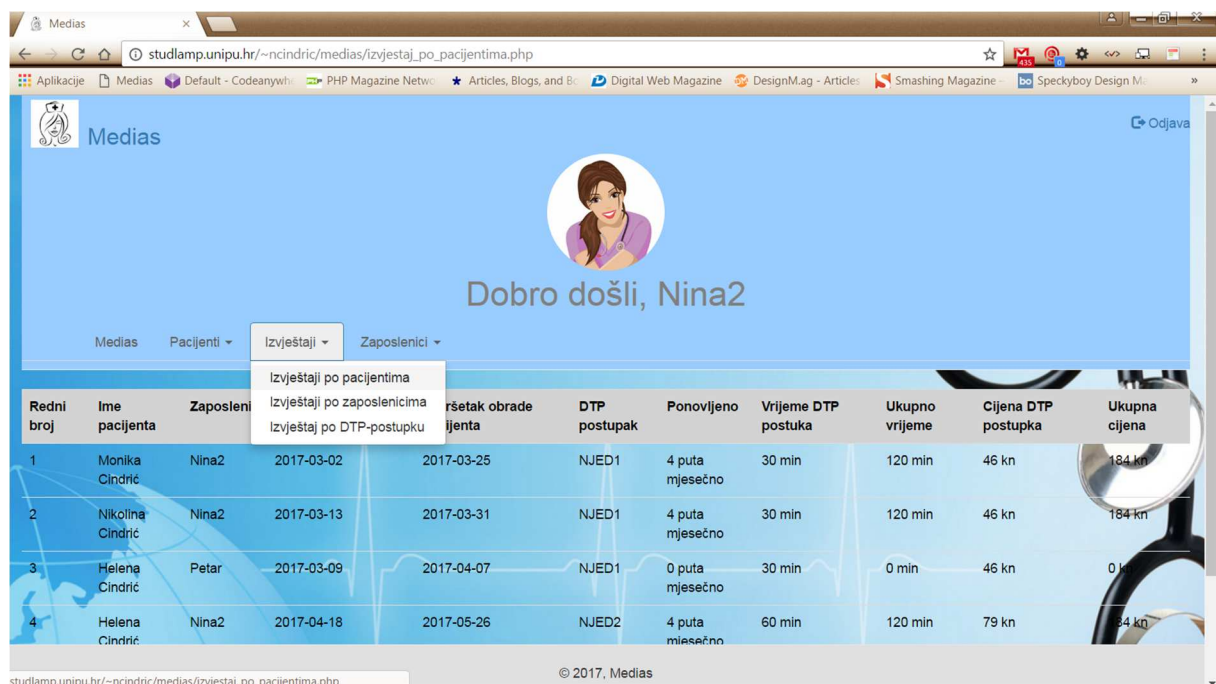


Slika 35: „Medias“ popis aktivnih pacijenata

izvor: izrada autora

5.2.9. Izvještaj po pacijentima

Link Izvještaji na navigacijskoj traci sastoji se od linkova izvještaji po pacijentima, izvještaji po zaposlenicima i izvještaji po DTP postupku. Stranica izvještaj po pacijentima sastoji se od zaglavlja i podnožja kao i sve stranice u sustavu aplikacije te glavnog dijela stranice koji se sastoji od tablice koja prikazuje podatke iz tablice „Obada“ i tablice „DTP“ po pojedinom pacijentu. U tablici su prikazani podatci o imenu i prezimenu pacijenta, zaposleniku, datumu početka i završetka obrade, DTP postupku, ukupnom mjesečnom broju posjeta zaposlenika pacijentu, pojedinačnom vremenu trajanja obrade i ukupnom mjesečnom vremenu trajanja obrade, jediničnoj cijeni DTP postupka i ukupnoj mjesečnoj cijeni DTP postupka. Podatci u ovoj tablici ne mogu se brisati ili mijenjati. Stranica izvještaja po pacijentima prikazana je na slici 36.



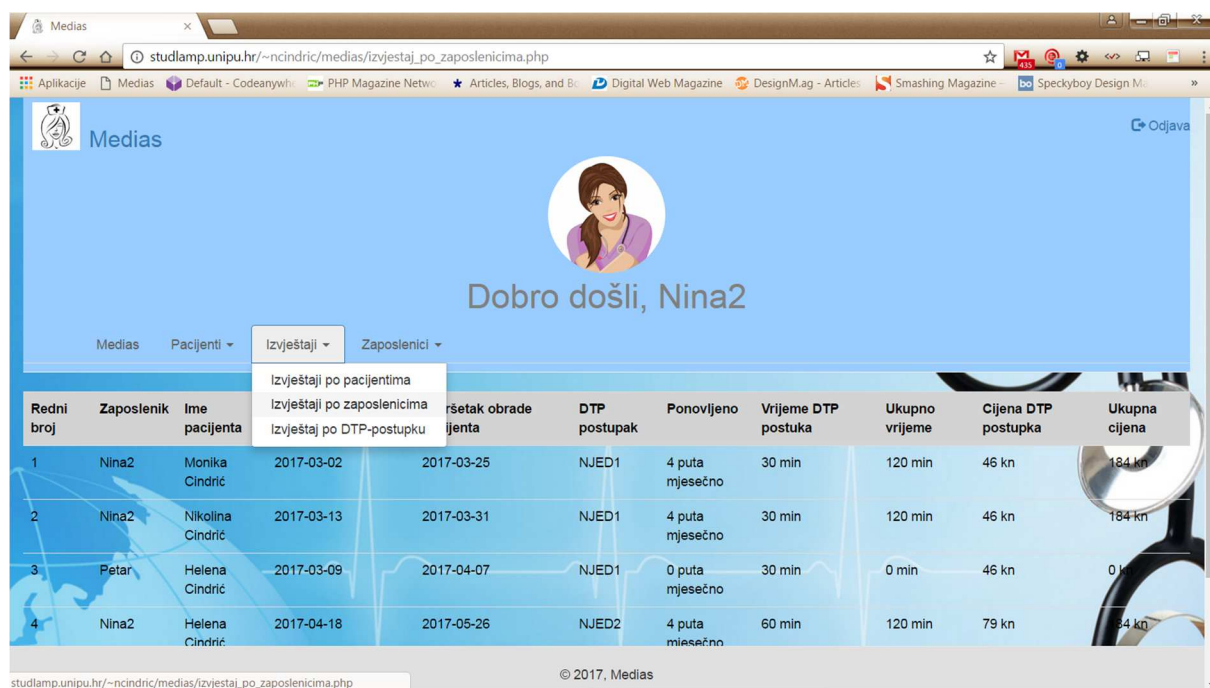
Redni broj	Ime pacijenta	Zaposleni	Izvještaj po pacijentima	Izvještaj po zaposlenicima	Izvještaj po DTP-postupku	Završetak obrade pacijenta	DTP postupak	Ponovljeno	Vrijeme DTP postupka	Ukupno vrijeme	Cijena DTP postupka	Ukupna cijena
1	Monika Cindrić	Nina2	2017-03-02			2017-03-25	NJED1	4 puta mjesečno	30 min	120 min	46 kn	184 kn
2	Nikolina Cindrić	Nina2	2017-03-13			2017-03-31	NJED1	4 puta mjesečno	30 min	120 min	46 kn	184 kn
3	Helena Cindrić	Petar	2017-03-09			2017-04-07	NJED1	0 puta mjesečno	30 min	0 min	46 kn	0 kn
4	Helena Cindrić	Nina2	2017-04-18			2017-05-26	NJED2	4 puta mjesečno	60 min	120 min	79 kn	316 kn

Slika 36: „Medias“ izvještaj po pacijentima

izvor: izrada autora

5.2.10. Izvještaj po zaposlenicima

Stranica izvještaj po zaposlenicima sastoji se od zaglavlja i podnožja kao i sve stranice u sustavu aplikacije te glavnog dijela stranice koji se sastoji od tablice koja prikazuje podatke iz tablice „Obada“ i tablice „DTP“ po pojedinom pacijentu poredano po zaposlenicima. U tablici su prikazani podatci o zaposleniku, imenu i prezimenu pacijenta kojega zaposlenik mora obraditi, datumu početka i završetka obrade, DTP postupku, ukupnom mjesečnom broju posjeta zaposlenika pacijentu, pojedinačnom vremenu trajanja obrade i ukupnom mjesečnom vremenu trajanja obrade, jediničnoj cijeni DTP postupka i ukupnoj mjesečnoj cijeni DTP postupka. Podatci u ovoj tablici ne mogu se brisati ili mijenjati. Stranica izvještaja po zaposlenicima prikazana je na slici 37.



Redni broj	Zaposlenik	Ime pacijenta	Izvještaj po zaposlenicima	Izvještaj po DTP-postupku	Početak obrade	DTP postupak	Ponovljeno	Vrijeme DTP postupka	Ukupno vrijeme	Cijena DTP postupka	Ukupna cijena
1	Nina2	Monika Cindrić	2017-03-02		2017-03-25	NJED1	4 puta mjesečno	30 min	120 min	46 kn	184 kn
2	Nina2	Nikolina Cindrić	2017-03-13		2017-03-31	NJED1	4 puta mjesečno	30 min	120 min	46 kn	184 kn
3	Petar	Helena Cindrić	2017-03-09		2017-04-07	NJED1	0 puta mjesečno	30 min	0 min	46 kn	0 kn
4	Nina2	Helena Cindrić	2017-04-18		2017-05-26	NJED2	4 puta mjesečno	60 min	120 min	79 kn	316 kn

Slika 37: „Medias“ izvještaj po zaposlenicima

izvor: izrada autora

5.2.11. Izvještaj po DTP-u

Stranica izvještaj po DTP-u sastoji se od zaglavlja i podnožja kao i sve stranice u sustavu aplikacije te glavnog dijela stranice koji se sastoji od tablice koja prikazuje podatke iz tablice „Obada“ i tablice „DTP“ po pojedinom pacijentu poredano po DTP-u. U tablici su prikazani podaci o DTP-u, pacijentu kojemu je potrebna obrada, zaposleniku koji je izvršio obradu, datumu početka i završetka obrade, ukupnom mjesečnom broju posjeta zaposlenika pacijentu, pojedinačnom vremenu trajanja obrade i ukupnom mjesečnom vremenu trajanja obrade, jediničnoj cijeni DTP postupka i ukupnoj mjesečnoj cijeni DTP postupka. Podatci u ovoj tablici ne mogu se brisati ili mijenjati. Stranica izvještaja po DTP-u prikazana je na slici 38.

Medias

studlamp.unipu.hr/~ncindric/medias/izvjestaj_po_DTP.php

Aplikacije

Medias

Default - Codeanywh

PHP Magazine Netw


Articles, Blogs, and B

Digital Web Magazine

DesignMag - Article

Smashing Magazine

Speckyboy Design M



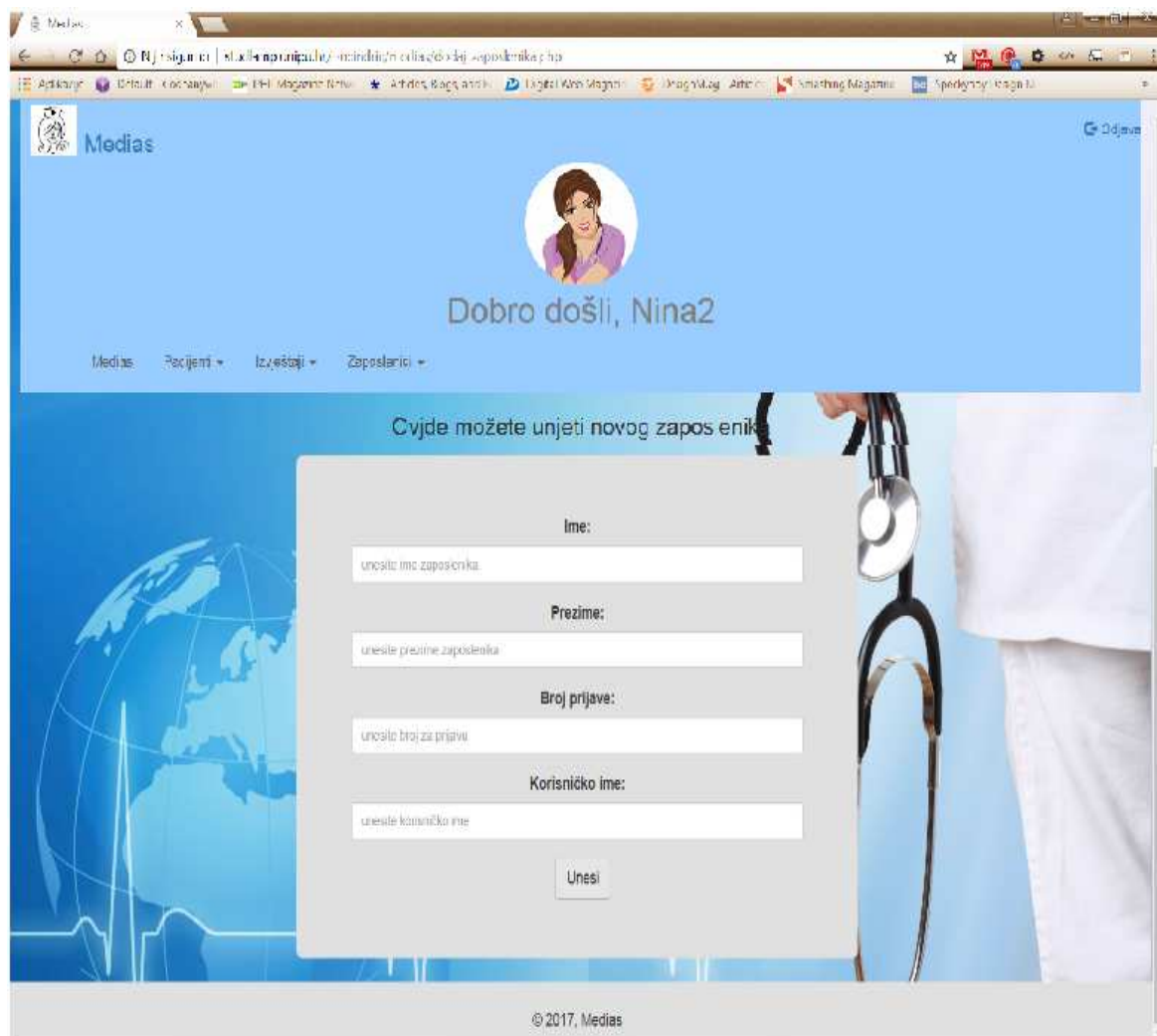
Medias

Slika 38: Medias izvještaj po DTP-u

izvor: izrada autora

5.2.12. *Unos zaposlenika*

Link zaposlenici na navigacijskoj traci sastoji se od linkova unos zaposlenika, popis zaposlenika i praćenje rada. Klikom na link unos zaposlenika dolazi se na stranicu koja omogućava unos novih zaposlenika u bazu podataka. Ova stranica sastoji se od zaglavlja i podnožja istog kao i sve stranice u sustavu „Medias“ te glavnog dijela stranice. U glavnom dijelu stranice nalazi se forma od osam polja od kojih su pet tekstualna polja, jedno polje padajućeg izbornika, jedno polje datuma i jedno password polje. Klikom na gumb unesi koristeći se jQuery Ajax-om i PHP-om uneseni podaci pohranjuju se u bazu podataka. Ako su podatci uneseni u bazu nakon klika na gumb unesi, javlja se upozorenje: „ Uspješno dodan novi zaposlenik!“ Ako podatci nisu uneseni u bazu podataka, javlja se upozorenje :„Ups! Zaposlenik nije dodan, molimo Vas da pokušate ponovno!“ Stranica unosa zaposlenika vidi se na slici 39.

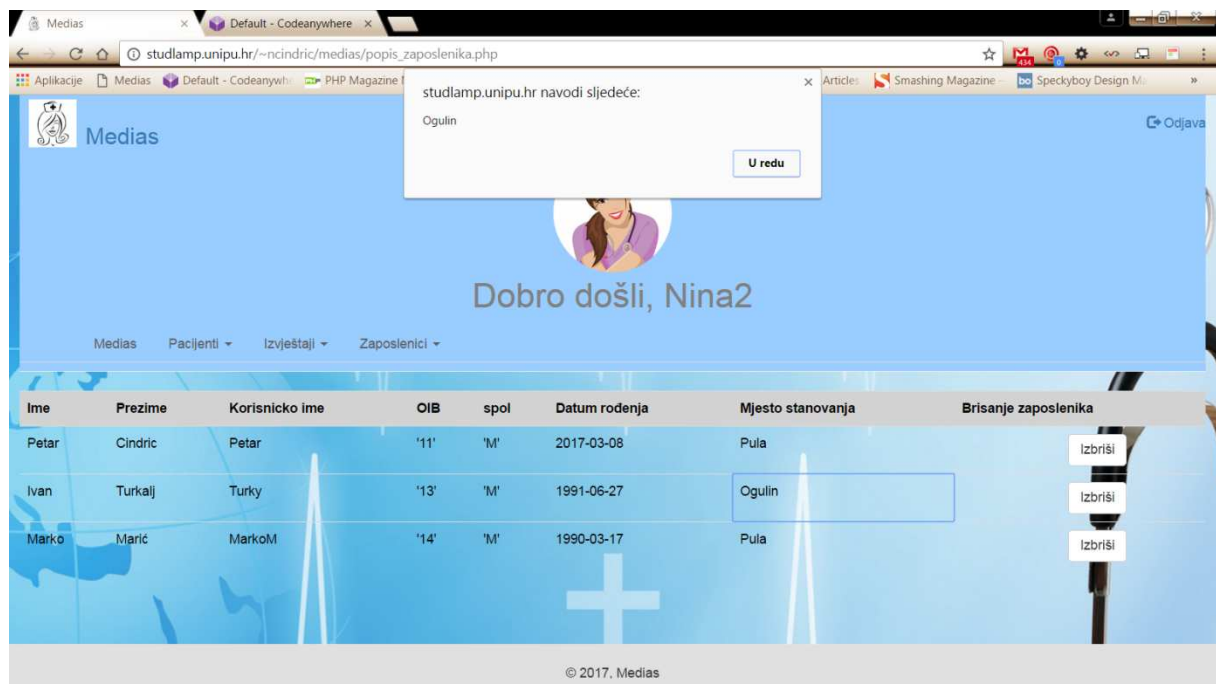


Slika 39: “Medias” unos zaposlenika

izvor: izrada autora

5.2.13. Popis zaposlenika

Stranica popis zaposlenik sastoji se od zaglavlja, podnožja te interaktivne tablice koja prikazuje podatke o imenu i prezimenu zaposlenika, korisničkom imenu, spolu, datumu rođenja i mjestu stanovanja zaposlenika pojedinog vlasnika ustanove. Tablica omogućava brisanje pojedinog zaposlenika iz baze podataka nakon potvrde o brisanju koje se javlja kao upozorenje, a nakon brisanja podataka iz baze podataka, tablica se automatski ažurira. Osim brisanja podataka iz baze podataka, tablica omogućava vlasnicima ustanove da promijene pojedine podatke o zaposleniku samim klikom na pojedinu ćeliju tablice koju žele promijeniti. Stranica popisa zaposlenika prikazana je na slici 40.

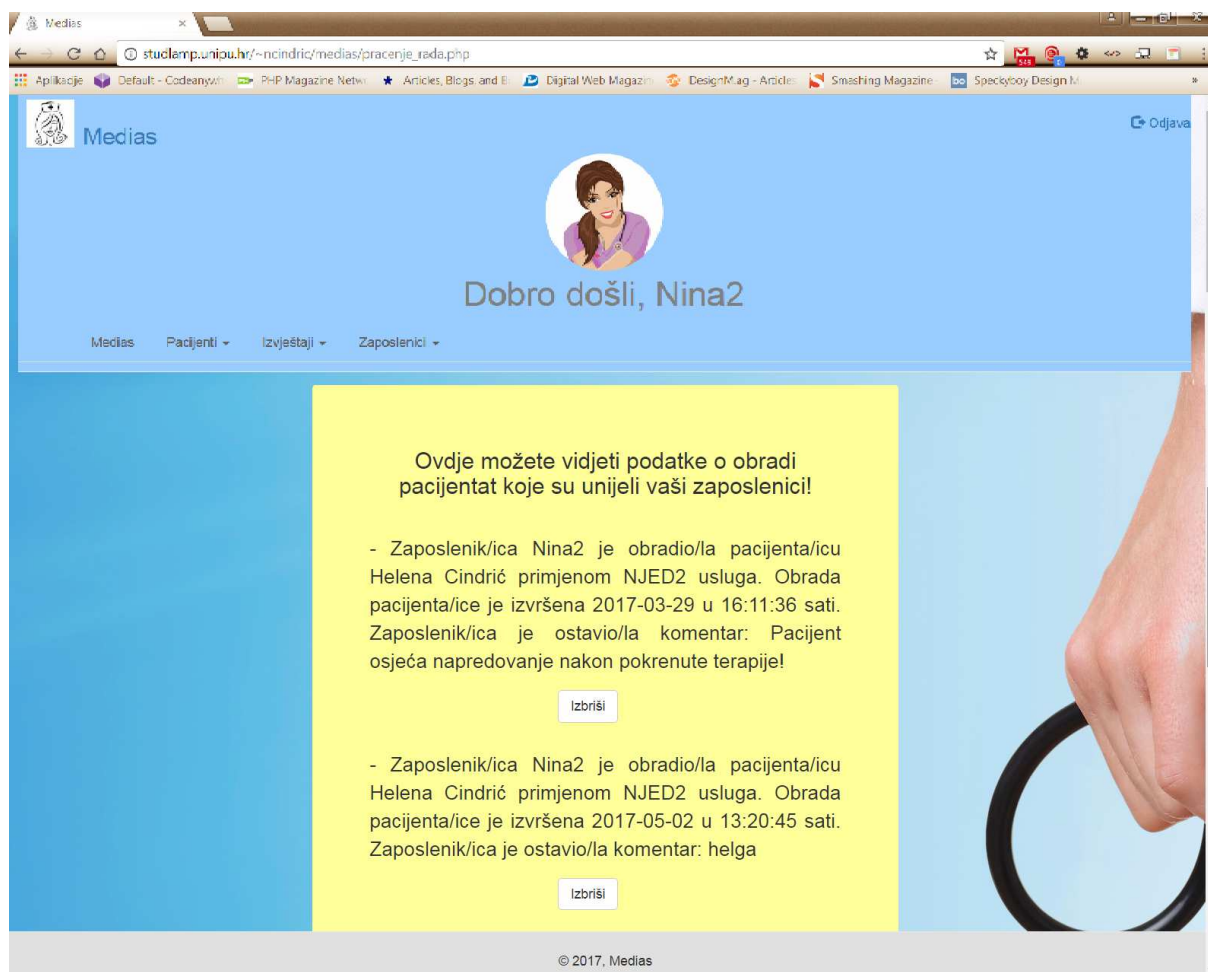


Slika 40: „Medias“ popis zaposlenika

izvor: izrada autora

5.2.14. Praćenje rada

Stranica praćenje rada omogućava vlasnicima ustanova pregled svih obavljenih obrada pacijenata s pripadajućim komentarima obrade koje su unijeli zaposlenici, a sastoji se od zaglavlja, podnožja te glavnog dijela s naslovom: „Ovdje možete vidjeti podatke o obradi pacijenata koje su unijeli vaši zaposlenici!“. Ispod naslova nalaze se podatci o obradi u sljedećim oblicima : - Zaposlenik - *korisničko ime zaposlenika koji je obradio pacijenta, ime i prezime pacijenta* primjenom DTP usluge. Obrada pacijenta je izvršena - *datum i vrijeme* i na kraju komentar kojeg unosi zaposlenik nakon obrade pacijenta. Podatci korisničko ime zaposlenika, ime i prezime pacijenta, naziv DTP-a, datum, vrijeme i komentar iščitani su iz tablice „Obrada_Gotova“ u bazi podataka. Vlasnik ima mogućnost brisati podatke iz baze podataka pritiskom na gumb Izbriši ispod pojedinog skupa podataka. Praćenje rada prikazano je na slici 41.

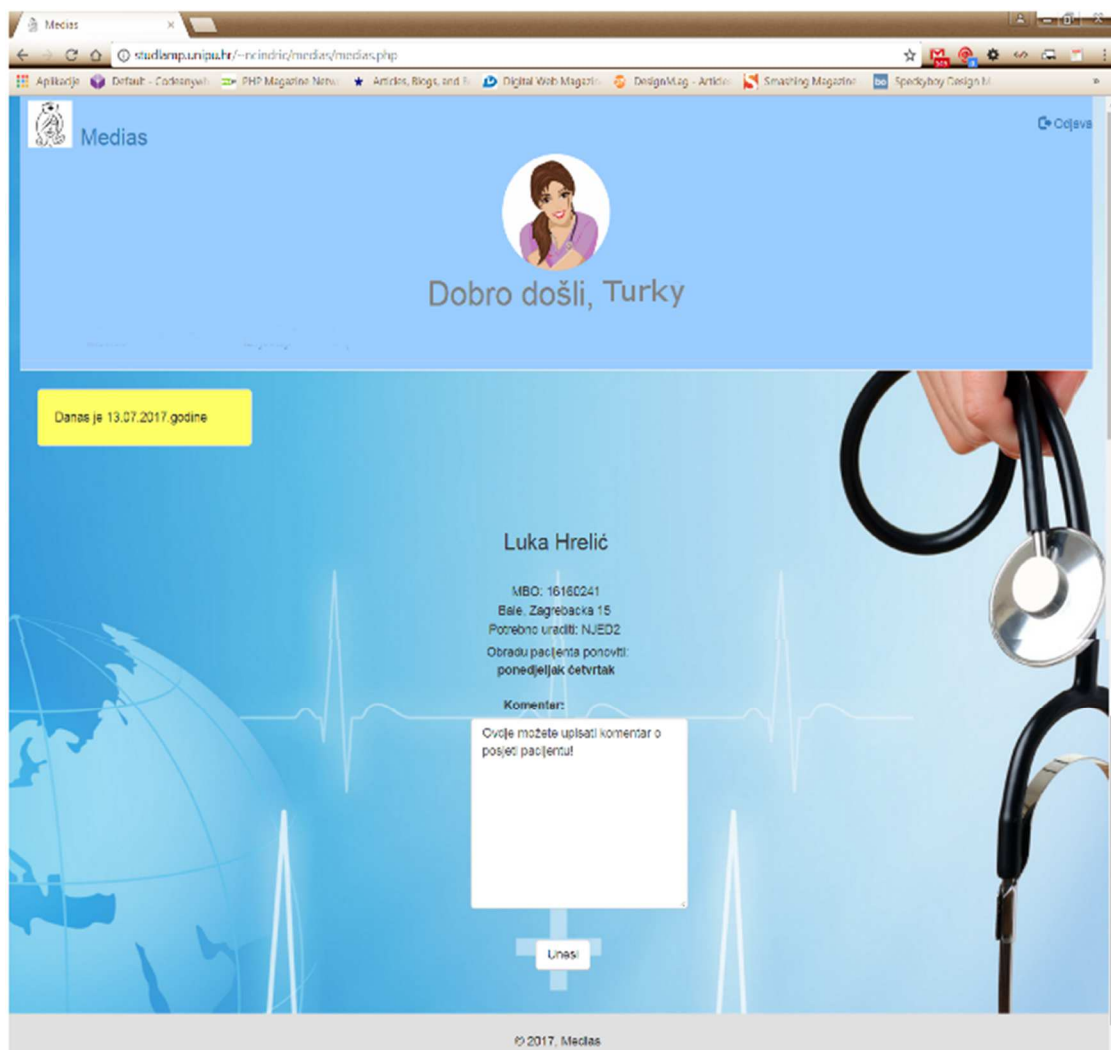


Slika 41: Medias praćenje rada

izvor: izrada autora

5.2.15. *MediasK*

Stranicu mediasK koriste zaposlenici ustanova za pružanje kućne njege nakon prijave u sustav na stranici Prijava popunjavanjem korisničkog imena i lozinke, odnosno broja za prijavu koje im definira vlasnik ustanove na stranici Unos zaposlenika. Podnožje i zaglavlje stranice jednako je kao i sva podnožja i zaglavlja nakon prijave u sustav „Medias“, no zaposlenici nakon prijave mogu pristupiti samo stranici mediasK koja nema navigaciju za druge stranice. Glavni dio stranice je isti kao i kod stranice medias kod vlasnika ustanove, odnosno sastoji se od pravokutnika u lijevom gornjem kutu u kojemu se nalaze vrijeme i sat. U sredini glavnog dijela nalazi se pravokutnik u kojemu su koristeći se PHP-om iz baze podataka izvučeni podatci o pacijentima koje prijavljeni zaposlenik treba obraditi odnosno ime, OIB, mjesto i adresa, njega koju pacijent treba te dan u tjednu kada pacijenta treba obraditi. Ispod ispisanih podataka o pacijentima, koji su aktivni za obradu, nalazi se polje u kojemu zaposlenik može upisati komentar o obradi pojedinog pacijenta, a koji se koristeći jQuery Ajax i PHP s prethodno navedenim podacima te vremenom i datumom šalju u bazu podataka, a vlasniku su vidljivi na stranici Praćenje rada. Stranica mediasK vidi se na slici 42.



Slika 42: MediasK

izvor: izrada autora

6. Zaključak

U diplomskome radu nastojala se prikazati izrada dinamičke web aplikacije „Medias“, koja je namijenjena vlasnicima ustanova za pružanje kućne njege pomoću koje mogu na jednom mjestu unositi podatke o svojim pacijentima i zaposlenicima, definirati obrade pojedinih pacijenata koje vrše pojedini zaposlenici, pratiti rad zaposlenika, pratiti pacijente koje sami moraju obraditi te na temelju izvještaja pratiti napredak pacijenata, ali i definirati osobne dohotke zaposlenika, korištenjem skriptnih jezika PHP i Ajax-a. Prilikom izrade same aplikacije korišten je evolucijski model razvoja softvera, tj. web aplikacije te je na temelju određenih karakteristika problema izrađena prvotna verzija aplikacije koja je prema potrebama vlasnika ustanove za kućnu njegu dorađivana do konačne verzije. Aplikacija je izrađena korištenjem web editora Codeanywhere te phpMyAdmin baze podataka koja se sastoji od šest tablica. Osim HTML kao osnovnog jezika, za izradu web stranica i aplikacija te za dizajn aplikacije korišten je CSS okvir Bootstrap, koji je omogućio prilagodbu aplikacije mobilnim uređajima, te CSS i JavaScript klijent-side skriptne jezike. Pozadinska strana aplikacije izrađena je koristeći server-side skriptni jezik PHP uz uporabu MySQL poziva na bazu podataka. Aplikacija se sastoji od 15 dijelova.

Nakon izrade dinamičke web aplikacije uporabom PHP-a i Ajax-a može se zaključiti kako uporaba ovih dvaju server-side skriptnih jezika prilikom izrade web aplikacija omogućava smanjenje vremena za prikaz traženih podataka ažuriranjem samo dijela stranice. Također, velika prednost uporabe ovih dvaju skriptnih jezika jest mogućnost slanja nekoliko upita prema serveru čega korisnici nisu niti svjesni. U procesu stvaranja aplikacije moglo se zaključiti kako uporaba PHP-a i Ajax-a prilikom izrade web stranice olakšava i ubrzava promet između preglednika i servera.

Literatura

Knjige:

1. Manger, R., *Softversko inženjerstvo*, Prirodoslovno-matematički fakultet u Zagrebu, 2005.
2. Božajić, I., *Uvod u PHP i objektno orijentirano programiranje*, Zagreb, Algebra učilište, 2011.
3. Lerdorf, R., *Programing PHP*, 2002.
4. Zakas, N.C., J. McPeak, J. Fawcett, *Profesional Ajax*, Indianapolis, Wiley Publishing, inc., 2006.
5. Crane, D., E. Pascarella, D. James, *Ajax in action*, Manning Publications, inc., 2006.

E-knjige:

1. Saternos, C., *Client-Server Web Applications with Javascript and Java: Rich, Scalable and Restful*, O'Reilly Media, inc, 2014., dostupno na: <https://books.google.hr/books?id=3b40AwAAQBAJ&printsec=frontcover&dq=web+applications+are+client-server&hl=hr&sa=X&ved=0ahUKEwjrbIJtorTAhWiKJoKHeLJAIYQ6AEIPTAF#v=onepage&q=web%20applications%20are%20client-server&f=false>, (pristupljeno: 5.4.2017.)
2. Shelly, G.B., Fraydenberg, M., *Web 2.0.: Concepts and Applications*, Cengage Learning, 2010., dostupno na: <https://books.google.hr/books?id=WYSP0Il6ftcC&printsec=frontcover&dq=web+2.0.&hl=hr&sa=X&ved=0ahUKEwjrtiriDzbzTAhUHQBoKHeFeApUQ6AEIQDAE#v=onepage&q=web%202.0.&f=false>, (pristupljeno : 26.4.2017.)
3. Spurlock, J., *Bootstrap: Responsive Web Development*, O'Reilly Media inc., 2013., dostupno na: <https://books.google.hr/books?id=LZm7Cxgi3aQC&printsec=frontcover&dq=web+applications+responsive+design+bootstrap&hl=hr&sa=X&ved=0ahUKEwjV4syNncLTAhVQrRQKHcisD2YQ6AEIZTAJ#v=onepage&q&f=false>, (pristupljeno: 26.4.2017.)
4. Stuttard, D. i M. Pinto, *The Web Application Hacker's Handbook: Finding and Exploiting Security Flaws*, 2. Izdanje, O'Reilly, John Wiley & Sons, 2011., Chapter 3: Web Applications Technologies, Web Functionality, dostupno na: <https://www.safaribooksonline.com/library/view/the-web-application/9781118026472/chap03-sec016.html>, (pristupljeno: 23.3.2017.)
5. York, R., *Beginning JavaScript and CSS Development with JQuery*, Indianapolis, Wiley Publishing, inc., 2009, dostupno na: <https://books.google.hr/books?id=L9otyT4crSQC&printsec=frontcover&dq=jquery&hl=hr&sa=X&ved=0ahUKEwibirjgroTAhUGWhQKHeuQAo0Q6AEIOTAE#v=onepage&q=jquery&f=false>, str. 19., pristupljeno (6.4.2017.)

6. Riordan, R., *Head First Ajax*, O'Reilly Media, Inc., 2008. godine dostupno na: <https://books.google.hr/books?id=sMMX-RWBbDkC&pg=PA9&dq=benefits+of++ajax&hl=hr&sa=X&ved=0ahUKEwiewOrNgpLTahUJP5oKHemNBdMQ6AEIHZA#v=onepage&q=benefits%20of%20%20ajax&f=false> , str. 2, (pristupljeno: 7.4.2017.)
7. Wald, K., J. Lengstorf, *Pro PHP and JQUERY*, dostupno na: <https://books.google.hr/books?id=PyWFCwAAQBAJ&pg=PA77&dq=jquery+ajax+request+on+php&hl=hr&sa=X&ved=0ahUKEwix3YfsjZLTahWHISwKHUmMAg0Q6AEIFzAA#v=onepage&q=jquery%20ajax%20request%20on%20php&f=false> , 2016., str. 77., (pristupljeno: 7.4.2017.)
8. Riordan, R., *Head First Ajax*, O'Reilly Media, Inc., 2008. godine dostupno na: <https://books.google.hr/books?id=sMMX-RWBbDkC&pg=PA9&dq=benefits+of++ajax&hl=hr&sa=X&ved=0ahUKEwiewOrNgpLTahUJP5oKHemNBdMQ6AEIHZA#v=onepage&q=benefits%20of%20%20ajax&f=false> , str. 10. , (pristupljeno: 7.4.2017.)

Članci – webstranice:

1. Stone Liver eLearning, "Top 5 Programming Languages Used In Web Development", dostupno na: <http://blog.stoneriverelearning.com/top-5-programming-languages-used-in-web-development/>, (pristupljeno: 23.3.2017.)
2. WordPress.org, History, dostupno na: <https://codex.wordpress.org/History> , (pristupljeno: 23.3.2017.)
3. Facebook for developers, Facebook SDK v5 for PHP, dostupno na: <https://developers.facebook.com/docs/reference/php/>, (pristupljeno: 23.3.2017.)
4. WikiMedia, dostupno na: <https://www.mediawiki.org/wiki/MediaWiki>, (pristupljeno: 23.3.2017.)
5. Federico Cargnelutti, "20 Most influential Open-Source Web Applications", dostupno na: <https://blog.fedecarg.com/2008/05/22/20-most-influential-open-source-web-applications/>, (pristupljeno: 26.4.2017.)
6. Joomla!, Documentation, dostupno na: https://docs.joomla.org/PHP_essentials, (pristupljeno: 26.4.2017.)
7. Coludways, „How PHP Will Fare IN 2017?“, dostupno na: <https://www.cloudways.com/blog/php-trends-2017/>, (pristupljeno: 24.3.2017.)
8. SpyreStudio, „Top Trending PHP Frameworks in 2016.“, dostupno na: <http://spyrestudios.com/top-php-frameworks-2016/>, (pristupljeno: 24.3.2017.)
9. PHP Manula, PDO, dostupno na: <http://php.net/manual/en/intro.pdo.php>, (pristupljeno 5.4.2017.)
10. PHP Manual, PDO Exception Class, dostupno na: <http://php.net/manual/en/class.pdoexception.php>, (pristupljeno: 5.4.2017.)
11. Segue Tehnologies, "Desktop vs. Web applications: A Deeper Look and Comparison", dostupno na: <http://www.seguetech.com/desktop-vs-web-applications/> , (pristupljeno: 5.4.2017.)

Prilozi

Sadržaj tablica:

Tablica 1: Prednosti i nedostaci web aplikacija	2
Tablica 2. Razlike između PHP okvira	21

Sadržaj slika:

Slika 1. Razlika između desktop aplikacija i web aplikacija	5
Slika 2. Prikaz korištenosti web aplikacija u odnosu na mobilne aplikacije	6
Slika 3. Dinamičke web aplikacije vs. statičke web stranice	7
Slika 4: Model vodopada	8
Slika 5: Model evolucijskog razvoja	9
Slika 6: Model inkrementalnog razvoja	10
Slika 7: Model formalnog razvoja	11
Slika 8: Model grafičkog razvoja	12
Slika 9. Model usmjeren na ponovnu uporabu	13
Slika 10: Klijent vs. server skriptni jezici	14
Slika 11: Korištenost PHP-a kao server-side skriptnog jezika	16
Slika 12: Najkorišteniji programski jezici	17
Slika 13: Najpopularniji PHP okviri u 2016. Godini	20
Slika 14: Konekcija na bazu podataka	22
Slika 15: Unos podataka u bazu podataka	23
Slika 16: Čitanje iz baze podataka	23
Slika 17: Brisanje podataka iz baze podataka	24
Slika 18: Izmjena podataka u bazi podataka	25
Slika 19: primjena \$.ajax() poziva	27
Slika 20: Prikaz podataka	28
Slika 21: \$.ajax() Izmjena podataka	29
Slika 22: Prikaz strukture tablice "Medias"	32
Slika 23: Prikaz strukture tablice "Pacijenti"	33
Slika 24: Prikaz strukture tablice "Zaposlenici"	33
Slika 25: Prikaz strukture tablice „DTP“	34
Slika 26: Prikaz strukture tablice „Obrada“	34
Slika 27: Prikaz strukture tablice „Obrada_Gotova“	35
Slika 28: Početna stranica Medias	36
Slika 29: „Medias“ registracija	37
Slika 30: „Medias“ prijava	38
Slika 31: Medias	40
Slika 32: Unos pacijenata	41
Slika 33: "Medias" obrada	42
Slika 34: „Medias“ popis pacijenata	43
Slika 35: „Medias“ popis aktivnih pacijenata	44
Slika 36: „Medias“ izvještaj po pacijentima	45
Slika 37: „Medias“ izvještaj po zaposlenicima	46

Slika 38: Medias izvještaj po DTP-u	47
Slika 39: “Medias” unos zaposlenika.....	49
Slika 40: „Medias“ popis zaposlenika.....	50
Slika 41: Medias praćenje rada.....	51
Slika 42: MediasK	53

Sažetak

Cilj ovoga diplomskog rada jest prikazati razvoj dinamičkih web aplikacija uporabom PHP-a i Ajax-a na stvarnoj aplikaciji. Rad je podijeljen u dva dijela koja pojašnjavaju sam proces razvoja dinamičke web aplikacije korištenjem navedenih skriptnih jezika. Prvi dio rada teorijski je dio u kojemu je pojašnjen pojam web aplikacija te njihova uloga u novom trendu WEB 2.0., nakon čega su prikazani modeli razvoja softverskog proizvoda, pa tako možemo razlikovati modele evolucijskog razvoja, inkrementalnog razvoja, formalnog razvoja, grafičkog razvoja te model usmjeren na ponovnu uporabu. U istom poglavlju uz opis pojedinog modela prikazane su i njihove prednosti i nedostaci. Slijedi poglavlje u kojemu su pojašnjene vrste skriptnih jezika, pa tako skriptne jezike možemo podijeliti na one usmjerene ka klijentu i one usmjerene ka serveru. Detaljno su pojašnjeni skriptni jezici koji se koriste prilikom razvoja web aplikacije u ovom diplomskom radu – PHP i jQuery Ajax. Prilikom opisa ovih dvaju server side skriptnih jezika prikazana je povijest njihova nastanka i razvoj njihova korištenja, statistički podatci o korištenosti PHP-a kao serverskog jezika, ali i novi trendovi i okviri u uporabi PHP-a te korištenje MySQL upita. Prilikom pojašnjavanja uporabe jQuery Ajaxa kao serverskog jezika korišteni su primjeri iz koda za izradu web aplikacije „Medias“. Drugi dio ovoga diplomskog rada primjena je stečenih saznanja na aplikaciji „Medias“, prilikom čega se prvo prikazala i pojasnila svaku tablicu u bazi podataka, koju sam koristila prilikom izrade web aplikacije, te nakon toga prikazala i pojasnila svaku od 15 stranica koje čine web aplikaciju „Medias“.

Ključne riječi: razvoj, dinamička web aplikacija, PHP, jQuery Ajax .

Summary

The aim of this graduate thesis is to show the development of dynamic web of the applications using PHP and Ajax on the actual application. The work is divided into two parts that explain the process of developing a dynamic web application using the above mentioned scripting languages. The first part of the work is a theoretical part in which concepts of web applications and their role in the new trend of WEB 2.0 are clarified. Thereafter, the models of software development are presented, so you can distinguish models of evolutionary development, incremental development, formal development, graphic development and re-use model. In the same chapter, along with the description of each model, both their advantages and disadvantages are shown. Furthermore, the types of scripting languages are clarified, so we can divide the scripting languages into those directed to the client and those directed to the server. This graduate thesis offers a detailed explanation of scripting languages used in developing web applications, of such as - PHP and jQuery Ajax. Their history of development and usage is shown, in the very description of these two server-side scripting languages, as well as statistics on PHP usage as a server language, including new trends and frames in the use of PHP and the use of MySQL queries. When clarifying the use of jQuery Ajax as a server language, examples from the code to create a web application Medias are used. The second part of this graduate thesis is application of acknowledgements in creating web application Medias, whereby is first showed and explained each database table which was used to create a web application, and than I presented and explained each of the 15 pages that are forming the Medias web application.

Keywords: Development, Dynamic Web Application, PHP, jQuery Ajax